

UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA



DIGITAL WORKPLACE - OFFICE 365 SOLUTION

Tiago Filipe Miranda Mota

MESTRADO EM ENGENHARIA INFORMÁTICA

Especialização em Sistemas de Informação

Trabalho de projeto orientado por:
Prof. Doutor Maria Isabel Alves Batalha Reis da Gama Nunes
e co-orientado por André Alves

2018

Agradecimentos

A realização deste trabalho de projeto contou com importantes apoios sem os quais não se teria tornado possível.

À minha namorada que está sempre presente, a apoiar-me em todas as alturas.

Aos meus pais que sem eles nunca chegaria onde cheguei, tenho de lhes agradecer por tudo e por todas as condições que sempre se sacrificaram para me dar.

Aos meus irmãos por todo o apoio que me deram e por me ajudarem sempre que precisei.

Aos meus avós pelo carinho e orgulho que sempre demonstraram ter.

À minha orientadora professora doutora Maria Nunes por todo o trabalho conjunto não só neste trabalho de projeto, mas ao longo de todo o percurso académico.

Ao meu orientador André Alves e à sua restante equipa de gestores de projeto, dos quais tenho de destacar a Inês Oliveira e o Joel Fonseca pela confiança que depositaram em mim e pela sua disponibilidade para me ajudarem nos períodos mais difíceis.

A todos os membros da Unipartner pelo excelente espírito de equipa que incorporam.

A todos os que me ajudaram direta ou indiretamente na realização do projeto de mestrado, o meu sincero obrigado.

Dedicatória.

Resumo

Com a utilização cada vez mais crescente das tecnologias/ferramentas que se encontram disponíveis na plataforma *Office 365*, são implementadas novas soluções de colaboração interna de *cloud-office* que tiram partido dessas mesmas funcionalidades, proporcionando uma melhor experiência de utilização aos utilizadores finais, facilitando a gestão de algumas operações implementadas no mundo empresarial.

Este relatório é composto por diversos capítulos que definem todo o processo inerente ao projeto desenvolvido, relacionando os mesmos de forma a encadear as ferramentas usadas e as funcionalidades desenvolvidas. Desenvolve os passos realizados na criação da *intranet* ao longo do projeto de estágio na empresa Unipartner IT Services S. A, que em futuras referências neste relatório será designada apenas como Unipartner. Será destacada a metodologia *Agile* que foi a metodologia utilizada na realização do projeto, assim como a motivação, âmbito e o propósito do mesmo. Vão ser destacados neste relatório os conceitos mais importantes, as tecnologias abordadas durante o projeto bem como as ferramentas da *Microsoft* que foram utilizadas por decisão do cliente da Unipartner, destacando-se entre as quais a tecnologia *SharePoint* e a ferramenta *Powell 365*. Por fim será descrito todo o trabalho realizado pelo aluno neste projeto objeto de relatório, assim como um tópico de discussão sobre os desenvolvimentos efetuados. Este relatório foi escrito com base no novo acordo ortográfico.

Palavras-chave: *Microsoft, Intranet, Agile, SharePoint, Powell 365*

Abstract

With the increasing use of the technologies /tools available on the Office 365 platform, new cloud-office collaboration solutions are deployed that take advantage of these same features providing a better user experience for end-users by overcoming some complexity and management issues in some business operations.

This report is composed of several chapters that define the entire process inherent to the developed project, relating them in a way to chain the tools used and the functionalities developed. It describes the steps taken in the creation of the intranet throughout the internship project in the company Unipartner IT Services S. A, which in future references in this report will be designated only as Unipartner. The Agile methodology, which was the chosen methodology for this project, will also be referred as well as the motivation, scope and purpose of the project. The most important concepts and the technologies covered during the project as well as the Microsoft tools that were used by decision of the client of Unipartner, among which the technology SharePoint and the tool Powell 365 will be highlighted in this report. Finally and most importantly the work done by the student in this reports project, as well as a discussion chapter on the developments carried out in the project will be presented. This report was written based on the new spelling agreement.

Keywords: Microsoft, Intranet, Agile, SharePoint, Powell 365

Conteúdo

Lista de Figuras	xiii
Lista de Tabelas	xv
Glossário / Índice Remissivo	xix
1 Introdução	1
1.1 Contexto	1
1.2 Unipartner	2
1.2.1 Equipa	2
1.3 Motivação	2
1.4 Objetivos	3
1.5 Contribuições	3
1.6 Estrutura do documento	3
2 Metodologias e Planeamento do projeto	5
2.1 <i>Agile</i>	5
2.1.1 Esquema da metodologia <i>Agile</i>	6
2.1.2 Méritos da metodologia <i>Agile</i>	6
2.1.3 Deméritos da metodologia <i>Agile</i>	7
2.2 Planeamento do projeto	7
2.2.1 Planeamento inicial	7
2.2.2 Alterações ao planeamento inicial	8
3 Tecnologias usadas	9
3.1 Trabalho relacionado	9
3.2 Vantagens do <i>cloud computing</i>	10
3.3 Problemas do <i>cloud computing</i>	10
3.4 <i>Microsoft</i>	10
3.5 <i>SharePoint Online</i>	11
3.5.1 <i>Content types</i>	11
3.5.2 Arquitetura do <i>Search</i>	11

3.6	<i>ShareGate</i>	14
3.7	<i>Powell 365</i>	15
3.7.1	Limitações <i>Powell 365</i>	17
3.7.2	Vantagens <i>Powell 365</i>	18
4	Arquitetura do projeto	21
4.1	<i>Digital Workplace</i>	21
4.2	Arquitetura funcional	21
4.2.1	Diagrama de contexto	21
4.2.2	Interfaces com outras aplicações e sistemas	23
4.3	Arquitetura técnica	24
4.3.1	Diagrama de arquitetura técnica	24
4.3.2	<i>SharePoint Online</i>	25
5	Trabalho Realizado	31
5.1	Desenvolvimentos e práticas comuns no projeto	31
5.1.1	Linguagens de programação	32
5.1.2	Criações e alterações a funcionalidades da <i>Microsoft</i>	33
5.2	Resultados de pesquisa na <i>peoplereults.aspx</i> e <i>servicesresults.aspx</i>	35
5.2.1	<i>Peoplereults.aspx</i>	35
5.2.2	<i>Servicesresults.aspx</i>	36
5.3	<i>Web parts</i> de taxas bancárias	37
5.3.1	<i>Calculated columns</i>	38
5.3.2	Mapeamentos de propriedades no <i>search schema</i>	39
5.4	<i>Web parts</i> do estilo de tabuladores	39
6	Discussão	41
6.1	<i>Performance</i>	41
6.2	Mapeamentos no <i>Search schema</i>	44
6.3	Aprovação documental	45
6.4	Modelo de permissões	46
6.5	Alternativas ao projeto e complementares	46
7	Conclusão	47
7.1	Comentários críticos	48
7.1.1	Ao produto	48
7.1.2	Ao projeto	49
7.2	Trabalho futuro	49
7.3	Alternativas ao <i>Powell 365</i>	50
	Bibliografia	52

Lista de Figuras

1.1	Logótipo da Unipartner	2
2.1	Metodologia Agile	6
2.2	Planeamento projeto	8
3.1	Arquitetura do modelo de <i>Search</i>	12
3.2	Esquema <i>Search schema</i> do cliente	14
3.3	Esquema <i>ShareGate</i> a migrar conteúdos	15
3.4	Esquema <i>Powell Manager</i>	16
3.5	Esquema <i>Site Collection</i>	17
4.1	Diagrama de contexto	22
4.2	Diagrama de arquitetura técnica	24
4.3	<i>SharePoint Online</i>	26
4.4	<i>User Profile</i>	26
4.5	<i>Graph API</i>	28
4.6	<i>Powell 365</i>	28
5.1	Detalhe de um <i>content type</i> criado no <i>Powell Manager</i>	34
5.2	Caraterísticas da propriedade <i>Title</i>	34
5.3	Vista da página <i>peoplereults.aspx</i>	36
5.4	Vista da página <i>servicesresults.aspx</i>	37
5.5	Página das taxas Euribor	38
5.6	Coluna <i>SharePoint</i> que devolve o dia de uma data do tipo <i>calculated column</i>	39
5.7	<i>Web part Tabs</i>	40
7.1	Exemplo de <i>website</i> desenvolvido em <i>SharePoint Online</i> para um empresa na área da construção civil	48

Lista de Tabelas

4.1	Componentes do sistema	23
4.2	<i>Interfaces</i> com outros sistemas	24
4.3	Componentes da arquitetura técnica	25

Glossário / Índice Remissivo

Active Directory O *Active Directory* é uma implementação de serviço de diretório no protocolo LDAP que armazena informações sobre objetos em rede de computadores e disponibiliza essas informações a utilizadores e administradores desta rede. 22, 25, 26, 35

cache hit É um estado em que os dados solicitados para processamento são encontrados em memória *cache*. 43

CDN A *Content Delivery Network* (CDN) é uma rede de distribuição de informação que permite fornecer conteúdo *web* de uma forma mais rápida a um grande número de utilizadores, distribuindo o conteúdo por múltiplos servidores de forma a efetuar a duplicação do mesmo e direcionar o conteúdo ao utilizador com base na proximidade do servidor. 24, 25, 28, 29, 44

check-in Processo que permite dar entrada a um ficheiro por parte de um único utilizador. 45

check-out Processo que permite que o utilizador liberte o ficheiro com as suas alterações. 45

client-side Refere-se aos desenvolvimentos efetuados no *front-end* da aplicação. 11, 41, 42, 44

cloud providers São os responsáveis por disponibilizar determinados serviços na *cloud* disponíveis para serem utilizados por terceiros. 9

content type Define, com a exatidão necessária, a informação do cliente em *fields*, isto é colunas *SharePoint*. 10, 11, 14, 17, 25, 29, 33, 34, 36, 49

Crawling *Crawling* é um método de pesquisa inteligente que decide que dados deverão ser retornados nos resultados de pesquisa. 12

data centers É um local onde estão concentrados os sistemas computacionais de uma organização, como um sistema de armazenamento de dados. 9

focus group Sessões de apresentação dos desenvolvimentos realizados a utilizadores-chave, realizadas em cada *release*. 31, 42

grid computing A *grid computing* é um modelo computacional capaz de alcançar uma alta taxa de processamento dividindo as tarefas entre diversas máquinas dispersas numa rede. 9

legacy Um sistema *legacy* é um sistema que se encontra desatualizado ou que precisa de ser substituído, para o qual estão disponíveis versões mais atuais. 1, 27, 43

masterpage A *masterpage* fornece a *interface* e o *layout* geral das páginas de um *site* em *SharePoint*. 25, 29, 42

out-of-the-box São funcionalidades disponibilizadas pelo produto sem ser necessário desenvolvimentos. 12, 45–49

owner É o proprietário de um determinado *site*, podendo alterar as definições do mesmo. 16

page layout É a estrutura da página *web*. 17, 18, 25, 31, 32

server-side Refere-se aos desenvolvimentos efetuados no *back-end* da aplicação. 11

site collection É um grupo de *sites* que têm o mesmo *owner* e que partilham as mesmas configurações. 14, 16, 17, 22, 23, 25, 28, 35, 44, 45

site structure É a estrutura do *website*, mostrando hierarquicamente os *subsites* que a constituem. 17

site template É uma estrutura aplicada ao nível dos *sites* de *SharePoint* que pode conter bibliotecas, listas, páginas, diferentes vistas, entre outros elementos. 17, 25

subsite É um *site SharePoint* com os mesmos privilégios que o *site* presente na *root*, estando definido no domínio URL após o *site* da *root* ou de outros *subsites*. 10, 16, 17, 25, 46

tenant Repositório *online*, no qual existem domínios *web*, utilizadores e grupos de segurança únicos. 15, 16, 25, 26, 28, 44, 45

theme É nesta estrutura que está inserido um ficheiro que comporta todo o código CSS referente à *site collection*. 10

third-party Um *software third-party* é um componente reutilizável que pode ser utilizado por entidades que não o desenvolveram. 25, 44

VM Uma *virtual machine* (VM) é uma emulação de um sistema de computador. 43

web part É um componente inserido nos *page layouts* que mostra as informações requisitadas pelo cliente. 7, 10, 17–19, 25, 28, 29, 31–40, 42–47, 49

wireframe É o esquema das páginas *web* realizado pelo *designer*, onde são indicadas as *web parts* que compõem a página. 31, 32

Capítulo 1

Introdução

Ao longo deste primeiro capítulo é descrito o contexto do projeto em estudo e, de uma forma breve, a empresa responsável pelo estágio, assim como a equipa onde fui inserido. Apresentam-se ainda a motivação, a contribuição e os objetivos do projeto. Por fim, é apresentada a estrutura do documento.

1.1 Contexto

Existem muitas e variadas empresas que mantêm o licenciamento da *Microsoft* há já alguns anos e que o tencionam continuar a manter. A *Microsoft*, à semelhança dos seus competidores, vai lançando para o mercado novas versões dos produtos afetos aos planos de licenciamento que as empresas utilizam. Por norma, por questões organizacionais, as empresas nem sempre tiram partido das versões mais recentes dos produtos da *Microsoft* que utilizam na sua área comercial. Porém, a *Microsoft* deixa de suportar versões mais antigas dos seus produtos, “pressionando” as empresas a adaptarem-se e, se pretendem manter o licenciamento, a utilizarem versões mais recentes, tornando as versões mais antigas numa espécie de versões *legacy* dos produtos da *Microsoft*.

O cliente da Unipartner situa a sua área de mercado no setor financeiro e, tal como as outras instituições desse mesmo setor, enfrenta uma série de desafios resultantes da instável dinâmica global e da acrescida pressão em matérias de regulação e controlo. Consciente deste contexto, o cliente da Unipartner estabeleceu linhas de orientação estratégicas no acompanhamento das tendências tecnológicas, tendo decidido apostar na renovação da sua *intranet* atual corporativa, desenvolvida em *SharePoint* 2007, versão *legacy* do *SharePoint* sobre a qual se desenvolviam soluções *web*, presente nos planos de licenciamento da *Microsoft*. Esta renovação tem como objetivo otimizar a aquisição e partilha de conhecimento entre os colaboradores.

A solução apresentada pela Unipartner é uma solução híbrida, que consiste numa migração funcional e tecnológica do conteúdo do cliente já existente para uma solução de *cloud-office*, que substituirá a *intranet* desenvolvida em *SharePoint* 2007 por uma nova

intranet desenvolvida em *SharePoint Online*; desta forma, tirará partido das ferramentas de produtividade do plano *Microsoft Office 365* juntamente com as novas funcionalidades a implementar requisitadas pelo cliente (abordadas mais à frente no relatório). Sobre a ferramenta *SharePoint Online*, o cliente da Unipartner comprou o licenciamento do produto *Powell 365*, um produto *Office 365*, que oferece várias opções de customização e permite obter uma melhor experiência de utilização da nova *intranet* do que o conseguido com *SharePoint Online* apenas.

1.2 Unipartner

A Unipartner foi criada em 2015 após a reorganização da Unisys que operava em Portugal. A Unipartner concede serviços de consultoria, de desenvolvimento aplicacional, de *cloud* e infraestruturas e gestão de projetos. É no cliente da Unipartner que se desenrola e se irá desenrolar a totalidade do projeto.



Figura 1.1: Logótipo da Unipartner

1.2.1 Equipa

Fui inserido numa equipa de projeto multidisciplinar, constituída inicialmente por um gestor de projeto, um *designer* e dois *developers*, onde me incluía. Dentro da equipa existiam diferentes níveis de experiência. Na fase inicial do projeto houve uma alteração na equipa, tendo saído da empresa um *developer* sénior, sendo substituído no imediato por um *developer* júnior. Posteriormente, entrou na equipa um colega sénior com experiência profissional nas várias versões de *SharePoint*. Os elementos da equipa já tinham experiência profissional nas ferramentas que foram usadas no projeto. O meu papel na equipa foi o de desenvolver novas funcionalidades e *layouts* da nova *intranet*, assim como na integração de funcionalidades já existentes requeridas pelo cliente usando o *Powell 365*, e também o *SharePoint Online*.

1.3 Motivação

Com vista a alcançar uma possível vantagem competitiva no mercado, um crescimento do seu produto bancário e fazer face a novos regulamentos de controlo, o cliente enfrenta agora um novo paradigma de atualização da sua *intranet*. Com este projeto pretende-se melhorar a experiência do utilizador e reduzir o *time-to-market* para o cliente, que

é significativo no modelo atual de *intranet*. Neste novo modelo de *intranet*, a mesma será constantemente atualizada com as novas funcionalidades e com os dados atuais e disponibilizada para os utilizadores interessados, corrigindo o desalinhamento da *intranet* atual face à realidade e às necessidades dos utilizadores.

1.4 Objetivos

O grande objetivo é a implementação de uma nova *intranet* em *SharePoint Online*, substituindo a *intranet* atual desenvolvida em *SharePoint 2007*, com ênfase nas seguintes características: rapidez; disponibilidade da informação pretendida; otimização da experiência de utilização; facilidade na gestão de conteúdo.

Para alcançar estes objetivos, teve-se como prioridades:

- não sobrecarregar o *website* de código cliente que pudesse vir a afetar a rapidez do mesmo;
- reestruturação dos conteúdos já existentes, acrescentando e alterando metadados de modo a que diversos conteúdos pudessem estar relacionados entre si;
- aposta em *layouts* inovadores, onde a grande maioria se centrava nas potencialidades do produto;
- desenvolvimento de novas funcionalidades que iam de encontro às necessidades dos utilizadores finais;
- adoção de um modelo de gestão que permita a atualização e evolução contínua da *intranet*.

1.5 Contribuições

Com este projeto foi desenvolvida uma nova *intranet* que permitirá potenciar a colaboração entre os seus colaboradores e a divulgação de eventos internos. Este projeto contribui para a contínua modernização tecnológica deste cliente, que continua a apostar nos produtos *Microsoft* como um fator vantajoso na sua atividade comercial.

1.6 Estrutura do documento

Este documento está organizado da seguinte forma:

- Capítulo 1 – Introdução - Trata-se do capítulo introdutório, onde se define o âmbito e os objetivos do projeto;

- Capítulo 2 – Metodologias e Planeamento - Enuncia as metodologias usadas no projeto, assim como o seu planeamento e os seus desvios;
- Capítulo 3 – Tecnologias usadas - São descritas as ferramentas e funcionalidades *Microsoft* usadas no projeto com relevância para o desenvolvimento do mesmo;
- Capítulo 4 – Arquitetura - É descrita a arquitetura do projeto, relacionando esquematicamente as ferramentas e funcionalidades da *Microsoft* usadas no projeto;
- Capítulo 5 – Trabalho realizado - É descrito o trabalho realizado no projeto, destacando-se as práticas comuns e desenvolvimentos *custom*;
- Capítulo 6 – Discussão - Neste capítulo referem-se os aspetos mais relevantes do projeto, assim como os obstáculos ao mesmo;
- Capítulo 7 – Conclusão - São apresentadas as conclusões finais do projeto, apresentando comentários críticos ao projeto bem como alternativas de ferramentas para o desenvolvimento de *intranets*.

Capítulo 2

Metodologias e Planeamento do projeto

Neste capítulo é descrita a metodologia de desenvolvimento de *software* que foi aplicada no projeto, assim como o planeamento do mesmo.

2.1 *Agile*

Neste projeto a metodologia de desenvolvimento de *software* foi a metodologia *Agile* [1]. Duas das principais características da metodologia *Agile* que tiveram uma maior relevância no projeto são a incremental e a iterativa. Incremental porque consiste num processo ágil em que não se tenta desenvolver o projeto todo de uma vez, em vez disso, são realizadas partições não-triviais do sistema que possibilitam o seu desenvolvimento incremental, sendo realizados testes individuais para cada partição. A partição só é integrada no sistema quando se encontra completa e aceite nos testes. Iterativa porque se baseia na construção do *software* em ciclos pequenos, definindo um conjunto de atividades. Para que uma determinada funcionalidade fique totalmente completa, este ciclo de atividades será repetido as vezes necessárias (iteração).

2.1.1 Esquema da metodologia Agile

Na figura 2.1 encontra-se descrita esquematicamente a metodologia Agile:

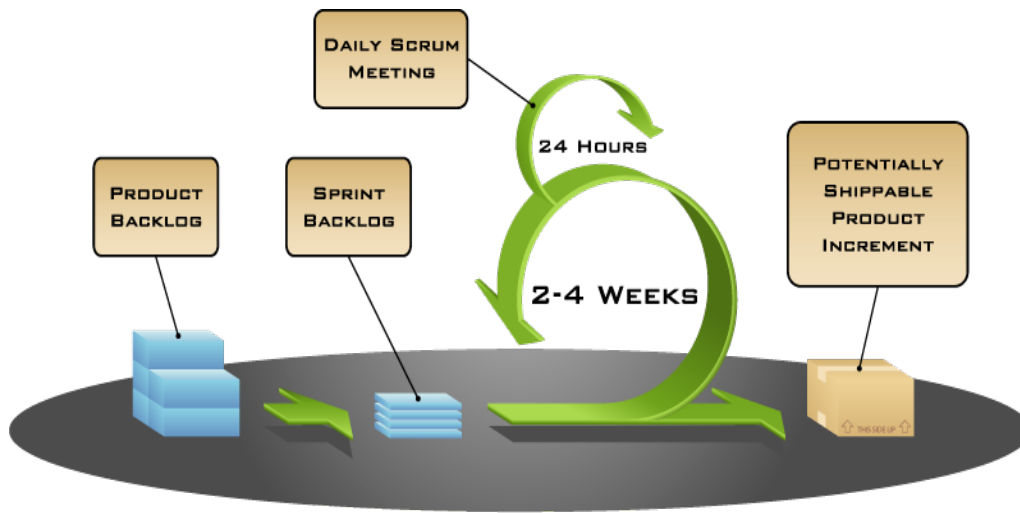


Figura 2.1: Metodologia Agile

Primeiramente os requisitos do produto são definidos, representados na imagem pelo “Product Backlog”. Consoante os requisitos, são planeadas as iterações necessárias para os desenvolver - os *sprints*. No início de cada *sprint* são realizadas reuniões para definir quais os requisitos a serem desenvolvidos, sendo que no início de cada dia do *sprint* também é realizada uma reunião mais pequena para se fazer um levantamento dos requisitos já realizados e de possíveis atrasos. O objetivo é, no final de cada iteração, ser possível incrementar o produto com os requisitos definidos para esse *sprint*. No final de cada *sprint* é realizada uma bateria de testes para validar o funcionamento esperado.

2.1.2 Méritos da metodologia Agile

Os principais méritos identificados pelo autor Charul Nigam [1] são:

- Melhoria na comunicação e coordenação da equipa de desenvolvimento;
- Com as constantes entregas dos *sprints*, os *developers* promovem a integração no sistema ao longo das semanas, em vez de ser ao longo dos meses ou anos;
- Com *sprints* mais curtos, permitem que o cliente forneça um *feedback* mais real sobre o projeto, possibilitando assim uma maior agilidade na possível alteração de requisitos, sem grande impacto no projeto;
- A metodologia *Agile* é dinâmica, permitindo que os projetos apresentem as suas especificidades, não os limitando de forma alguma.

2.1.3 Deméritos da metodologia *Agile*

Os principais deméritos identificados pelo autor Charul Nigam [1] são:

- Exige que as competências técnicas dos *developers* estejam aprimoradas, não se podendo especializar numa determinada competência mas sim num leque de competências mais alargado;
- Torna a gestão da equipa mais complexa e moldável;
- A gestão entre equipas que desempenham diferentes papeis em projetos dependentes - se alguma destas equipas não implementar uma metodologia ágil, podem surgir dificuldades no agendamento das entregas.

2.2 Planeamento do projeto

2.2.1 Planeamento inicial

O Projeto de Engenharia Informática (PEI) iniciou-se no dia 1 de Outubro de 2017. No período antes de se iniciar o projeto alvo de relatório, contactei com ferramentas que iria usar no futuro. O projeto alvo de estágio iniciou-se no dia 20 de Novembro de 2017 e tinha como data de fecho dia 17 Maio de 2018. É subdividido em três fases principais, que por sua vez são compostas por várias *milestones*. Estas três fases principais encontram-se descritas como a formulação de três portais:

- A partir do portal *myCEMG*, que tem como objetivo final tornar-se na área pessoal dos utilizadores, os mesmos poderão aceder a conteúdos, tais como o *email*, eventos, tarefas, entre outros. Para além de outros desenvolvimentos, para este portal criei duas páginas *web*, nas quais desenvolvi *web parts* que retornam resultados de pesquisa centrados nos colaboradores e nos departamentos da empresa. Este desenvolvimento será abordado no capítulo do trabalho realizado;
- O portal *vNext* nesta nova *intranet* colaborativa consiste num novo paradigma na experiência do utilizador, com a reorganização dos conteúdos, a reformulação do *layout* e uma navegação e funcionalidades *responsive*. Para além de outros desenvolvimentos, para este portal desenvolvi várias *web parts* que utilizam uma lógica e têm a aparência de tabuladores. Este desenvolvimento será abordado no capítulo do trabalho realizado;
- O portal *MyCEMG vNext* será uma área partilhada entre os colaboradores onde se irão desenvolver funcionalidades à medida do cliente que promovam a colaboração dos trabalhadores da empresa. Para além de outros desenvolvimentos para esta última fase, desenvolvi *web parts* que retornam valores de taxas bancárias usadas

no negócio do cliente. Este desenvolvimento será abordado no capítulo do trabalho realizado.

Diagrama do planeamento

Fase 1 - Portal *myCEMG*

- Protótipo - 20 Dezembro
- *Focus Group* - 3 a 4 Janeiro
- Testes Aceitação - 17 a 23 Janeiro
- Entrada em produção - 24 Janeiro

Fase 2 - Portal *vNext*

- Protótipo– 30 Janeiro
- *Focus Group* - 9 a 14 Fevereiro
- Testes Aceitação – 12 a 16 Março
- Entrada em produção – 19 Março

Fase 3 - Portal *MyCEMG vNext*

- *Focus Group* – 28 março a 3 Abril
- Testes Aceitação – 8 a 14 Maio
- Entrada em produção – 17 Maio

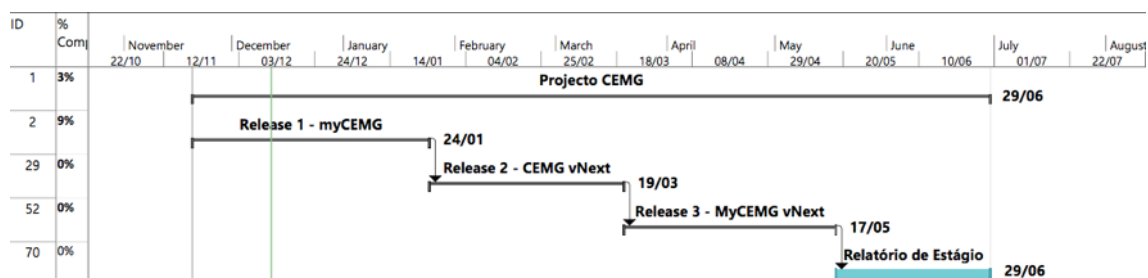


Figura 2.2: Planeamento projeto

2.2.2 Alterações ao planeamento inicial

O planeamento inicial sofreu alterações de requisitos e de prioridades por parte do cliente após a entrada em produção da Fase 1. Estas alterações fizeram prolongar a entrada em produção das Fases 2 e 3 do projeto, prolongando a data fim do projeto para 17 de Junho de 2018. As entregas das Fases 2 e 3 ocorreram dentro do novo planeamento do projeto.

Capítulo 3

Tecnologias usadas

Neste capítulo começa-se por abordar alguns temas relacionados com o projeto, de onde se destaca a importância da *cloud computing*, nomeadamente através do uso da ferramenta *Microsoft SharePoint Online*. São também referidas ferramentas da *Microsoft* que tiveram uma grande relevância no desenvolvimento da *intranet*, como é o caso da ferramenta *Powell 365* e a ferramenta *ShareGate*, assim como as linguagens de programação usadas.

3.1 Trabalho relacionado

O *SharePoint Online*, que é a ferramenta *Microsoft* com a qual foi realizado o projeto em estudo, centra-se na área das novas tecnologias de *cloud computing*. O artigo [2] explora o impacto nas entidades organizacionais das novas tecnologias, nomeadamente a *cloud computing* e a *Web 2.0*.

Desde que a *cloud computing* emergiu em 2007, gerou um grande interesse tecnológico junto dos seus interessados (autores, consultores, analistas e organizações). À medida que o interesse aumentava, houve várias tentativas para definir *cloud computing*. Um estudo [2] realizado por McKinsey “The Global Management consulting firm”, descobriu que existiam cerca de 22 possíveis definições para *cloud computing*. Grossman (2009) e Voas Zhang (2009), afirmam mesmo que não parece existir uma definição ou um *standard* para *cloud computing*.

Sultan e Sultan (2012) tentaram definir de forma unânime *cloud computing*. Segundo os autores, *cloud computing* é uma modalidade que usa os avanços das tecnologias de informação e comunicação (TIC), como a virtualização e *grid computing*, para fornecer uma variedade de serviços de TIC através de *softwares* e *hardwares* virtuais fornecidos por organizações *cloud providers* que detêm os próprios *data centers*. Os requisitos/serviços disponibilizados aos utilizadores são distribuídos tanto por redes públicas como por privadas, havendo até modelos de distribuição híbrida de serviços TIC. Estes serviços TIC: incluem *softwares* desenvolvidos que vão de encontro às necessidades das organizações; fornecem capacidades de processamento rápidas e quase ilimitadas e

possuem uma capacidade de armazenamento quase ilimitado; fornecem ferramentas que permitem aos seus clientes desenvolver e gerir as suas próprias aplicações.

3.2 Vantagens do *cloud computing*

Quando a *cloud computing* surgiu em 2007, vários analistas tiveram diferentes reações: enquanto que uns viram méritos outros, entre eles várias personalidades respeitadas como o fundador da *Oracle*, consideraram-na um modelo de negócio inútil. Apesar destas opiniões díspares, esta nova tecnologia continuou a atrair vários seguidores, o que levou a que várias organizações a adotassem e comesçassem a oferecer os seus serviços via *cloud*.

Passado algum tempo após a implementação da *cloud computing*, já poucos analistas tinham dúvidas sobre o impacto económico positivo deste novo paradigma, que oferece uma grande variedade de *softwares* e *hardwares*, permitindo escalabilidade e evitando a necessidade de fazer grandes investimentos em licenças de *softwares* e *hardwares* reduzindo custos significativos para as organizações.

3.3 Problemas do *cloud computing*

Apesar da flexibilidade e do potencial económico demonstrado pela *cloud computing* ainda existem muitos problemas a ultrapassar, nomeadamente problemas de segurança, problemas internos dos fornecedores e interrupções dos seus serviços, que são os mais problemáticos. A segurança é, sem dúvida, uma das principais preocupações das organizações. O cliente para o qual se realizou o objeto de estudo não foge à regra, tendo sempre demonstrado uma grande preocupação com os dados da organização.

Um inquérito [2] realizado a 244 diretores de informação e executivos realizado em 2008 pela IDC (*International Data Corporation*), revelou que 75% dos inquiridos classificaram a segurança como a sua principal preocupação com a *cloud computing*, enquanto que o desempenho e a disponibilidade eram as outras duas preocupações para 63% dos inquiridos.

3.4 Microsoft

A *Microsoft* apresentou em 2011 uma nova plataforma, o *Office 365*, sustentada em infraestruturas de *cloud*. Este *cloud-based software* denominado de *SharePoint Online*, que pertence aos produtos de *software* disponíveis na plataforma *Office 365*, possibilita aos utilizadores a criação e a publicação de *websites* sem grande complexidade de programação, tendo apenas de modificar componentes como *themes*, *templates*, *subsites*, *web parts* (também denominados de *widgets*) e elementos de estrutura de dados (*content types*).

Dependendo da complexidade dos *websites* que se pretendem desenvolver em *SharePoint Online*, podem-se associar ferramentas como o *SharePoint Designer* e/ou o *Visual Studio* (usado na maioria dos *websites* que contêm *server-side*; neste projeto existe apenas *client-side*). À semelhança da grande maioria dos produtos da *Microsoft*, que comunicam entre si, também o *SharePoint Online* tem essa capacidade de ligar vários componentes, o que traz uma mais-valia aos clientes que têm o seu negócio montado em estruturas da *Microsoft*.

Num fórum económico de reputação mundial, Neelie Kroes, vice-presidente da comissão Europeia para a tecnologia, destacou a grande importância da *cloud computing* para a economia, fazendo a seguinte afirmação: “Cloud Computing will change our economy. It can bring significant productivity benefits to all, right through to the smallest companies, and also to individuals. It promises scalable, secure services for greater efficiency, greater flexibility, and lower cost. . .”

3.5 *SharePoint Online*

As funcionalidades disponibilizadas pelo *SharePoint Online* que tiveram um papel importante no desenvolvimento do *website* em estudo e que suscitaram algumas questões por parte do cliente sobre o produto, foram as designadas abaixo.

3.5.1 *Content types*

Uma organização com relativa importância no seu mercado alvo possui diferentes tipos de informação que, todos juntos, traduzem toda a informação útil da mesma. Apesar de as informações das organizações terem quase sempre propriedades em comum, cada tipo de conteúdo terá as suas particularidades e poderá ser partilhado/disponibilizado pela organização de maneiras diferentes, ou mesmo utilizar diferentes políticas de confidencialidade dos dados.

Como existia esta necessidade de organização de dados, foi definido, há já algumas versões anteriores do *SharePoint*, o conceito de *content type*. Um *content type* [3] é um grupo de configurações reutilizáveis que descrevem os atributos e comportamentos partilhados para um determinado tipo de informação.

Os *content types* fornecem às organizações uma maneira de gerir e organizar o seu conteúdo de uma forma consistente em *sites*, listas e bibliotecas, além de possibilitar que uma lista ou biblioteca tenha vários tipos de *content types* associados.

3.5.2 *Arquitetura do Search*

O *Search* é o mecanismo de pesquisa usado pela ferramenta *SharePoint* para devolver aos utilizadores todo o conteúdo corporativo armazenado na *intranet*.

Após a consolidação de fatores de grande importância, como é o caso da *performance* e da escalabilidade, a *Microsoft* adotou para os seus produtos uma nova arquitetura para o *search* [4], chamada de “*Fast and SharePoint Search*”, apresentada na figura 3.1. Esta arquitetura de *search* consiste num modelo estrutural definido pela *Microsoft*, para *websites* construídos em *SharePoint*, que permite pesquisas que devolvem conteúdos com exatidão e eficiência.

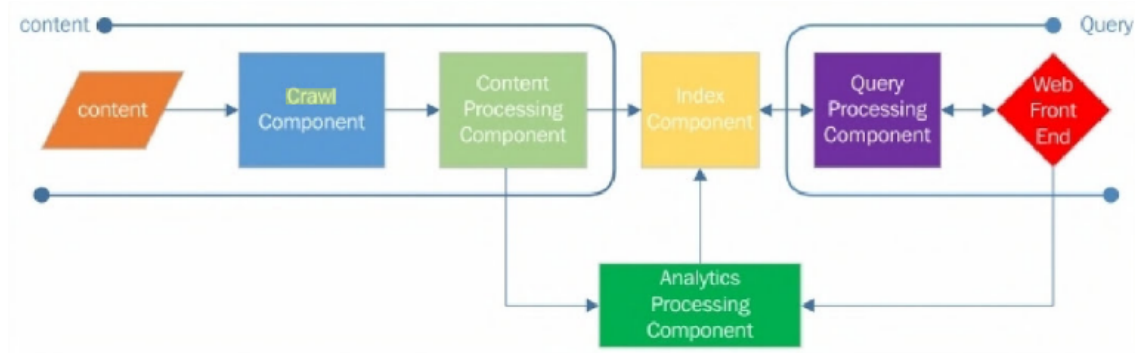


Figura 3.1: Arquitetura do modelo de *Search*

Como se pode observar na figura 3.1, esta arquitetura pode ser dividida em 4 componentes principais: *Content*, *Query*, *Index* e processamento analítico. Um dos assuntos mais abordados pelo cliente durante a realização do projeto, centrou-se na funcionalidade *out-of-the-box* do *SharePoint*, o *Crawl*, que se encontra na secção do *Content*. Como tal, o mesmo irá ser analisado em pormenor.

Crawl

Tal como ilustra a figura 3.1, o componente principal designado como *Content* é constituído por três elementos fundamentais.

O primeiro elemento é precisamente o conteúdo, isto é, a informação da organização, que se pretende que seja indexada de modo a que a sua consulta fique disponível. O segundo elemento consiste no *Crawl*, que é o elemento responsável pelo *Crawling* (método de pesquisa inteligente que decide que dados deverão de ser retornados nos resultados de pesquisa) dos conteúdos. O *Crawl* invoca conetores, *out-of-the-box* ou customizados, que interagem com a informação de modo a que esta seja disponibilizada. Enquanto ocorre a indexação no *Crawl*, são recolhidos metadados que a irão definir. Quando o *Crawl* termina, entrega estes dados ao último elemento do componente *Content*, designado por *Content-processing Component*. De uma forma geral, este último elemento tem como função receber do *Crawl* os dados e disponibilizá-los ao *Index Component*. Toda a informação presente no *Index Component* está disponível para a consulta/observação do utilizador final.

As “queixas” do cliente centravam-se nestes três passos descritos acima, que compõem o componente *Content*. Uma vez que foi usada a versão *Online* do *SharePoint* para este projeto, numa pesquisa tem de se esperar a cada 15 minutos que o *continuous crawler* ocorra e indexe novos ficheiros, ou alterações aos existentes. Como o cliente na sua antiga *intranet* usava a versão *on-premise* 2007 do *SharePoint* o *Crawl* ocorria no momento, uma vez que os dados ficavam guardados localmente, ao invés do que acontece no *continuous crawler* do *SharePoint Online* onde os dados são armazenados na *cloud*. Na secção da conclusão deste relatório será apresentado um exemplo prático de como o tempo exagerado do *continuous crawler* dificulta a validação da introdução correta dos conteúdos na *intranet*.

Search center

Trata-se de um centro de pesquisa [5] disponível onde os utilizadores do produto *SharePoint* podem realizar as suas pesquisas, sendo para isso disponibilizado um *site* no seguinte domínio `<hostName>/search/`. Este *site* terá uma página de resultados de pesquisa predefinida, podendo conter mais páginas de resultados de pesquisa onde podem ser devolvidos conteúdos mais específicos.

Search schema

Quando um utilizador pesquisa por conteúdos em *sites SharePoint*, são-lhe devolvidos apenas os itens que se encontram indexados na pesquisa, tal como ilustra a figura 3.1 no *Web Front End* na secção da *Query*. Os utilizadores só veem os resultados de pesquisa, consoante se têm ou não permissões de leitura para os mesmos.

O conteúdo e os metadados são chamados de propriedades dos itens. O *search schema* [6] controla os conteúdos que os utilizadores podem observar nos *websites*. O *search schema* é composto por uma lista de *crawled properties* que facilitam ao *crawler* devolver o conteúdo e os metadados. Porém, nem todo o *crawled content* ou os metadados são úteis para existirem no *search index*. O *search schema* possui então uma lista que contém todos os conteúdos úteis, designada por *managed property*. O *index* apenas inclui os conteúdos mapeados na *managed property*.

Isto significa que uma pesquisa simples por uma determinada palavra irá retornar os itens que contêm essa palavra nas suas propriedades mapeadas. Se se pretender pesquisar por uma determinada propriedade basta apenas escrever o nome da *managed property* seguido da palavra específica, como por exemplo, “author:James”.

Na figura 3.2 apresenta-se um exemplo da vista de administração do *search schema* do cliente do projeto em estudo, onde se podem observar algumas propriedades existentes, bem como as suas características e o seu mapeamento.

Administração de Coleções de Sites - Propriedades Geridas

Propriedades Geridas | Propriedades Pesquisadas | Categorias

1-50 | ▶

Utilize esta página para ver, criar ou modificar propriedades geridas e mapear propriedades pesquisadas para propriedades geridas. A pesquisa extrai automaticamente as propriedades pesquisadas dos conteúdos pesquisados. Para adicionar os conteúdos de uma propriedade pesquisada ao índice de pesquisa, mapeie a propriedade pesquisada a uma propriedade gerida. Pode utilizar as definições da propriedade gerida para limitar os resultados de pesquisa. A pesquisa cria automaticamente propriedades geridas para colunas de sites que contêm valores. As propriedades geridas criadas automaticamente são apresentadas nesta página a texto cinzento. As alterações a uma propriedade gerida entrarão em vigor após a pesquisa completa seguinte. Tenha em conta que as definições que podem ser ajustadas dependem do seu nível atual de autorização.

Filtro

Propriedade gerida: 

 Nova Propriedade Gerida

Nome da Propriedade	Tipo	Vários	Consulta	Pesquisar	Obter	Refinar	Ordenar	Seguro	Propriedades Pesquisadas Mapeadas	Aliases
RefinableDate00	Data e Hora	Vários	Consulta	-	Obter	Refinar	Ordenar	Seguro	ows_q_DATE_PublishingStartDate	
RefinableDate01	Data e Hora	Vários	Consulta	-	Obter	Refinar	Ordenar	Seguro	ows_q_DATE_ArticleStartDate	
RefinableDate02	Data e Hora	Vários	Consulta	-	Obter	Refinar	Ordenar	Seguro	ows_DueDate	
RefinableDate03	Data e Hora	Vários	Consulta	-	Obter	Refinar	Ordenar	Seguro	ows_q_DATE_EventDate	
RefinableDate04	Data e Hora	Vários	Consulta	-	Obter	Refinar	Ordenar	Seguro	ows_q_DATE_EndDate	
RefinableDate05	Data e Hora	Vários	Consulta	-	Obter	Refinar	Ordenar	Seguro	ows_q_DATE_Intranet_DataDeArquivo	
RefinableDate06	Data e Hora	Vários	Consulta	-	Obter	Refinar	Ordenar	Seguro	LastModifiedTime	
RefinableDate07	Data e Hora	Vários	Consulta	-	Obter	Refinar	Ordenar	Seguro	ows_Intranet_DataLimite	
RefinableDate08	Data e Hora	Vários	Consulta	-	Obter	Refinar	Ordenar	Seguro	ows_Intranet_DataInicio	

Figura 3.2: Esquema *Search schema* do cliente

3.6 ShareGate

Outra ferramenta da *Microsoft* utilizada no projeto foi o *ShareGate* que foi o produto responsável pelas migrações do conteúdo da *intranet* antiga para a nova. Na maioria dos conteúdos migrados o formato da informação manteve-se, nomeadamente para os ficheiros PDF que continuaram no formato PDF. No entanto, houve alguns conteúdos em formato de página (.aspx) cujas informações, aquando da migração, passaram a ficar disponíveis em formato PDF. No *ShareGate* tem de se definir o URL de origem e o URL de destino, podendo-se forçar valores para determinados campos dos *content types* das estruturas disponíveis no URL destino.

Esta ferramenta foi ainda usada para a migração de conteúdos entre os ambientes da nova *intranet*. Teve o papel de acelerador, visto que não se tinha de criar novamente os mesmos conteúdos para uma *site collection* diferente.

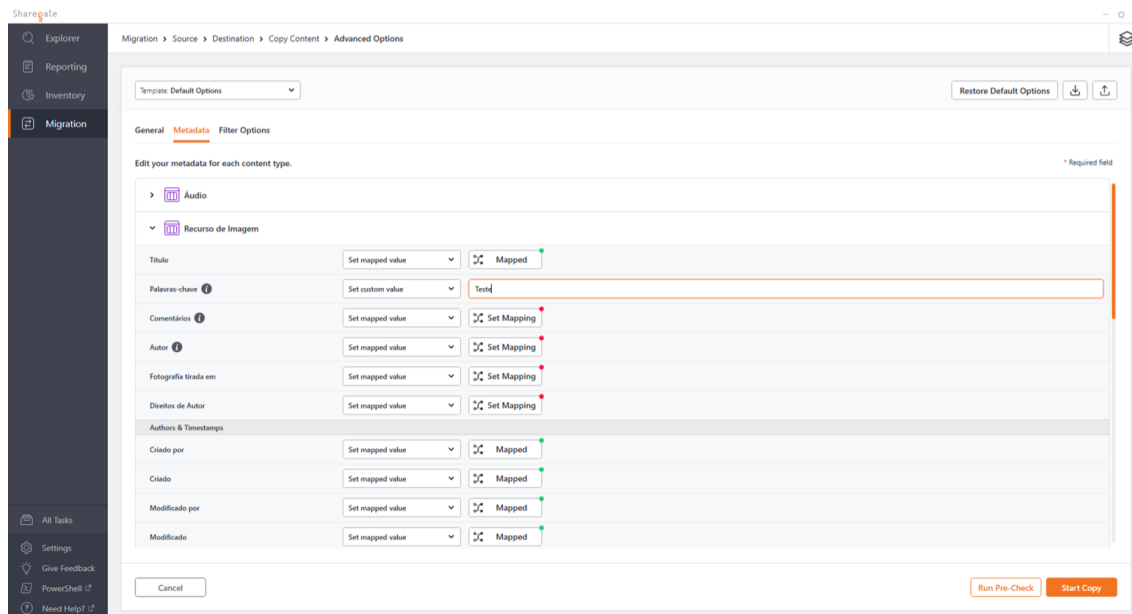


Figura 3.3: Esquema *ShareGate* a migrar conteúdos

Na figura 3.3 pode-se observar o mapeamento dos campos existentes numa lista *SharePoint* de um ambiente origem para um ambiente destino. Se os campos existirem em ambos os ambientes, o mapeamento é feito de forma automática, como é o caso do campo “Título”. Para outros campos que não existam em ambos os ambientes, terá de se definir o mapeamento a realizar, tal como acontece para o campo “Palavras-chave”.

3.7 *Powell 365*

No projeto objeto de estudo, fiquei responsável pelo desenvolvimento praticamente integral ao nível da ferramenta *Powell 365*. A função desta ferramenta para este projeto foi a de melhorar o aspeto gráfico que os *websites* desenvolvidos apenas em *SharePoint Online* apresentam. O *Powell 365* foi a ferramenta com maior importância no desenvolvimento da *intranet*, é um *software* relativamente recente lançado pela *Microsoft* em 2015, de modo a ir de encontro a uma das necessidades conhecidas dos utilizadores do *Sharepoint*, que é o facto dos *websites* desenvolvidos em *SharePoint* não terem o visual mais apelativo e a sua *user experience* não ser a ideal, sendo um *software* mais conhecido pela sua fiabilidade.

O cliente foi então aconselhado pela Unipartner e por representantes da *Microsoft Portugal*, a utilizar a ferramenta *Powell 365* na construção da sua nova *intranet*. Foi no *Powell Manager* que foi desenvolvida a maior parte da *intranet*, tendo sido feitas apenas pequenas customizações a nível do *SharePoint*.

No *Powell Manager*, ilustrado na figura 3.4, foi criado um *tenant* para o projeto, ou

seja, um repositório *online* onde é possível a criação de contas no ambiente *Microsoft Online Services*, no qual existem domínios *web*, utilizadores e grupos de segurança únicos. O *tenant* [7] pode conter várias instâncias disponíveis no produto *Office 365*, como foi para este projeto, onde se criaram várias *site collections* em *SharePoint Online*.

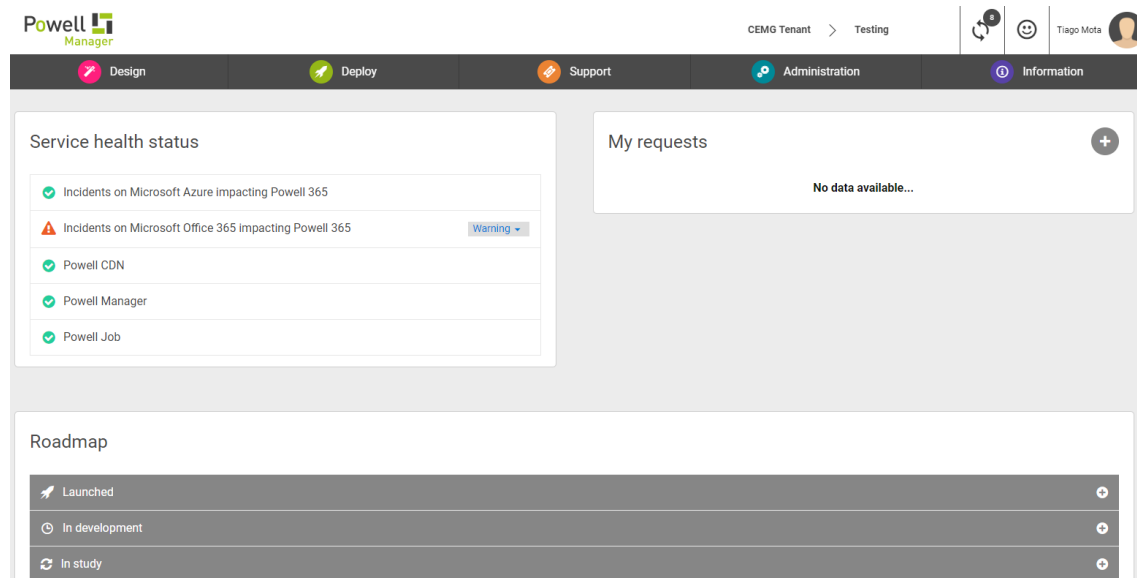
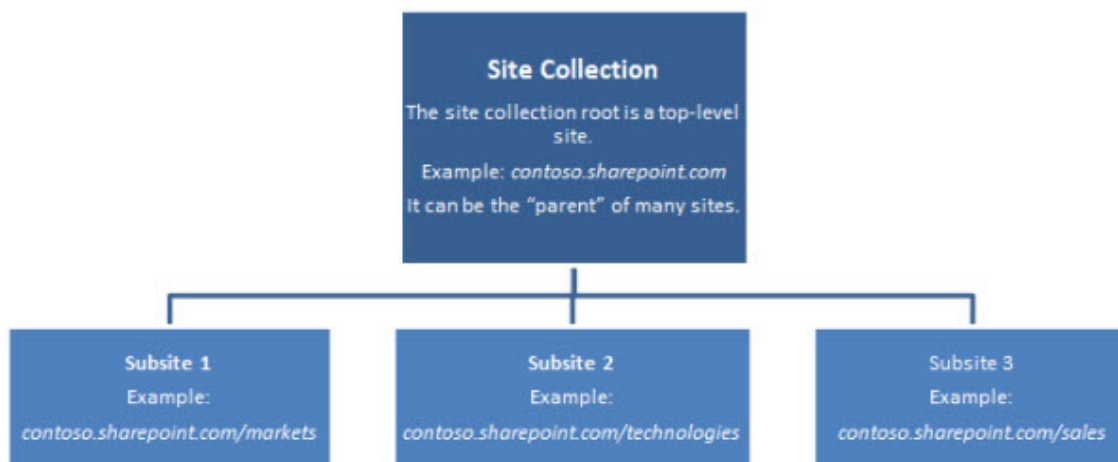


Figura 3.4: Esquema *Powell Manager*

Uma *site collection* [8] é um grupo de *sites* que têm o mesmo *owner* e que partilham as mesmas configurações, das quais se destaca a partilha das permissões. Quando é criada uma *site collection*, é criado automaticamente um *site* de nível superior, podendo-se criar sobre este *site* vários *subsites*. Podem-se definir novas permissões, assim como outras configurações, tais como a criação de grupos de *SharePoint*, específicos para cada *subsite*, sendo útil para organizações que não pretendem que todo o *site* seja visível para todos os seus colaboradores (sendo visíveis apenas os *subsites* para os quais existe um interesse funcional).

Figura 3.5: Esquema *Site Collection*

Para cada *release* foram definidos os ambientes disponíveis, nomeadamente ambientes de desenvolvimento, de qualidade e um ambiente único de produção. Cada um destes ambientes representa uma *site collection*.

Ao nível de cada *site collection* é definida a *site structure*, que consiste na definição hierárquica dos *subsites*, tal como ilustra a figura 3.5, sendo comum para todos os seus ambientes definidos.

Para cada *subsite* é aplicado um *site template* [9], que consiste numa estrutura que pode conter bibliotecas, listas, páginas, diferentes vistas, entre outros elementos, podendo-se reutilizar este *site template* para vários *subsites*. Ao nível dos *site templates* são definidas as páginas existentes no *subsite*, assim como as bibliotecas de documentos ou imagens, como a possível introdução de novos conteúdos que enriquecem o *website*, na secção dos *available page layouts*. Para cada página definida no *site template* é definido um *page layout*, que é a estrutura da página, e para cada secção do *page layout* existe a possibilidade de integrar uma ou mais *web parts*.

Aos *available page layouts* definidos para a introdução de conteúdos por parte do utilizador final, são associados *content types* que podem ser criados à medida. Às listas e às bibliotecas são também associados *content types*.

3.7.1 Limitações *Powell 365*

Após desenvolver um *website* colaborativo com base na ferramenta *Powell 365*, senti que facilita o desenvolvimento de *websites* deste âmbito. No entanto, apesar de ser um acelerador no desenvolvimento de projetos de *web development*, deparei-me com algumas limitações.

Controlo de versões

O *Powell 365* não tem controlo de versões ao contrário das versões mais recentes do *SharePoint*, onde para se consultar o histórico de versões basta aceder ao item pretendido e seleccionar a opção “histórico de versões”. Ao seleccionar cada versão é disponibilizado o estado do item nessa versão havendo a possibilidade de tornar essa versão a versão principal.

Como o *Powell 365* não tem controlo de versões, para qualquer funcionalidade disponível a versão anterior é substituída pela versão mais recente, não havendo a capacidade de restaurar versões anteriores, o que complicou por vezes o desenvolvimento de certas *web parts* e *page layouts*.

O *Powell 365* disponibiliza uma funcionalidade para os conteúdos que funciona como uma alternativa ao versionamento, que é a possibilidade de clonar as estruturas. Num caso prático onde pretendíamos manter duas versões sobre o mesmo objeto com pequenas alterações, fizemos uso da funcionalidade do *clone*, criando um novo objeto independente do objeto à imagem do qual foi criado, sobre o qual realizámos as pequenas alterações que pretendíamos. Obviamente que no final do projeto criámos muitos mais objetos, que apenas serviram como temporários, do que necessitaríamos se houvesse um controlo de versionamento.

Trabalho em paralelo

A não existência de versionamento nos objetos criados no *Powell 365* dificulta o trabalho em paralelo sobre o mesmo objeto. No *Powell 365* não existe o conceito de *token*, nem o conceito de *merge* ao nível do objeto. Se a versão mais recente do objeto não for igual à versão que se quer gravar, esta nova versão não irá conter as alterações da versão anterior.

Autenticação para as sincronizações

Quando se muda de ambiente, ou para se realizar novos desenvolvimentos ou para se fazer passagem de desenvolvimentos entre ambientes, a autenticação, que é definida ao nível da configuração dos ambientes, é perdida. Tem que se voltar às definições do ambiente de desenvolvimento em questão e reintroduzir as credenciais.

3.7.2 Vantagens *Powell 365*

As grandes vantagens do uso da ferramenta *Powell 365* para o desenvolvimento de *websites* construídos sobre *SharePoint Online* são as destacadas abaixo, que possibilitam a construção rápida de *websites* colaborativos.

Reutilização das estruturas

A reutilização das estruturas desenvolvidas no *Powell 365* são uma mais-valia, dado que se pode ter como ponto de partida estruturas gerais disponibilizadas pelo produto. Por exemplo, uma *web part* genérica fornecida pelo produto, sobre a qual se realizam alterações para apenas disponibilizar a informação pretendida pelo cliente.

Facilidade de aprendizagem

Por ser uma ferramenta disponível no mercado há relativamente pouco tempo, vai de encontro às expectativas dos aceleradores de projetos, não sendo necessário um grande tempo de adaptação para um desenvolvimento *web* nesta ferramenta. Prova disso foi o facto de ter sido o primeiro projeto na Unipartner a usar esta ferramenta e de ter sido concluído nos tempos delineados no projeto.

Capítulo 4

Arquitetura do projeto

Relativamente à arquitetura da *intranet*, designada como *Digital Workplace*, trata-se de uma plataforma totalmente desenvolvida em *Office 365*, tendo apenas como dependência a receção de dados externos que alimentam alguns componentes.

4.1 *Digital Workplace*

O *Digital Workplace* é um *website* interno composto pelas diversas áreas relevantes para o cliente que, agregadas, se afirmam como a definição interna do cliente. As principais áreas de negócio são: a secção documental, que é composta por documentos e por comunicações internas; a secção dos colaboradores, que é composta por diversas páginas que possibilitam a colaboração entre empregados; a secção de produtos financeiros, que é composta por produtos financeiros em vigor e por todas as partes de negócio relacionadas, como é o caso das taxas bancárias.

4.2 Arquitetura funcional

Nesta secção são descritas as funcionalidades implementadas na aplicação. Aqui são descritas as principais fronteiras da aplicação, os *inputs* e os *outputs*, e nas restantes são descritos os requisitos funcionais.

4.2.1 Diagrama de contexto

O esquema na figura 4.1 apresenta o diagrama de contexto onde é possível identificar os componentes que comunicam com o *Digital Workplace*.

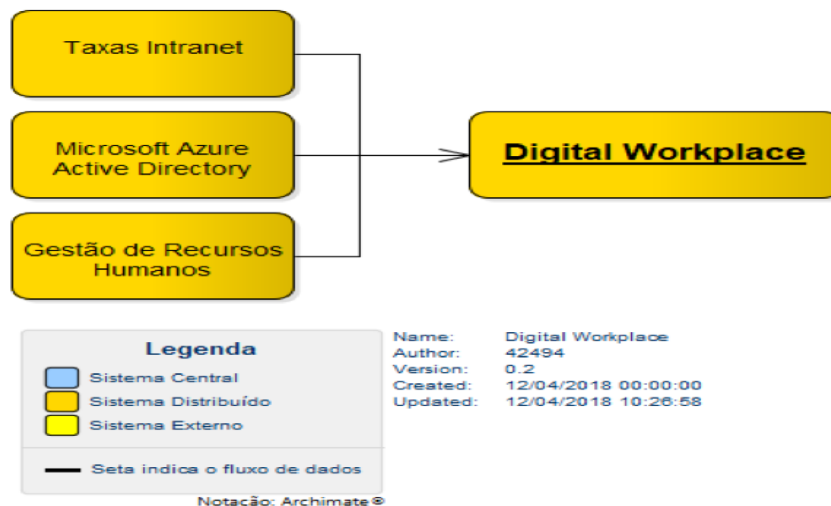


Figura 4.1: Diagrama de contexto

As “Taxas Intranet” são taxas bancárias cujos dados são fornecidos pelo cliente via *webservice* que popula listas *SharePoint* presentes na *site collection*. A “*Microsoft Azure Active Directory*” possui toda a informação dos departamentos e empregados da empresa, “alimentando” o componente *GesVen-GRH* e *User Profile* respetivamente. A “Gestão de Recursos Humanos” fornece documentos relevantes através do repositório *One Drive*. O *Yammer* é uma plataforma social que permite a comunicação entre colaboradores. A *Delve* permite a edição das propriedades dos perfis dos utilizadores e é ainda disponibilizado um canal de vídeos para a *intranet*. Estes três componentes gerais fornecem informação externa relevante para a constituição do *Digital Workplace*. Na tabela 4.1 que se segue são descritos os componentes do sistema.

APLICAÇÃO / COMPONENTE	DESCRIÇÃO
Taxas bancárias	Para que a informação sobre taxas bancárias possa ser mostrada no <i>Digital Workplace</i> , é passada diariamente esta informação para o <i>Digital Workplace</i> via <i>webservices</i> .
GesVen-GRH	Para que a informação institucional seja consumida no <i>Digital Workplace</i> , é passada diariamente informação departamental para o <i>Digital Workplace</i> via <i>webservices</i> .
<i>User Profile</i>	Permite que a informação relativa a utilizadores seja consumida no <i>Digital Workplace</i> por via do serviço de <i>User Profile</i> do <i>Office 365</i> .
<i>One Drive</i>	Permite que cada utilizador tenha uma <i>site collection</i> privada onde pode aprovisionar os seus documentos e se assim o desejar sincronizá-los com o seu posto de trabalho.
<i>Yammer</i>	Plataforma social profissional que permite interação social com todos os membros do portal.
<i>Delve</i>	Permite a visualização do perfil de cada utilizador, bem como a atualização de algumas das suas propriedades.
Vídeo	Permite que sejam criados canais de vídeo onde podem ser carregados e seguidos vídeos de acordo com as permissões.

Tabela 4.1: Componentes do sistema

4.2.2 Interfaces com outras aplicações e sistemas

As *interfaces* definidas permitem que sejam passadas para o *Digital Workplace* informações sobre taxas, bem como informação sobre a hierarquia da organização. Na tabela 4.2 são descritos, para os componentes descritos na tabela 4.1, os sistemas de origem e destino, assim como o modo em que os componentes são disponibilizados à *intranet*.

SISTEMA ORIGEM	SISTEMA DESTINO	TIPO	ONLINE / BATCH
Taxas	Digital Workplace	WebServices	Batch
GesVen-GRH	Digital Workplace	WebServices	Batch
User Profile	Digital Workplace	Serviço O365	Online
One Drive	Digital Workplace	Serviço O365	Online
Yammer	Digital Workplace	Serviço O365	Online
Delve	Digital Workplace	Serviço O365	Online
Video	Digital Workplace	Serviço O365	Online

Tabela 4.2: Interfaces com outros sistemas

4.3 Arquitetura técnica

O produto *Office 365* disponibiliza um vasto leque de aplicações, no entanto nem todas estão a ser utilizadas no contexto do projecto do *Digital Workplace*. Também no contexto desta aplicação são utilizados componentes do *Powell 365*, mais concretamente *scripts* provisionados na CDN (*Content Delivery Networks*) *Powell 365*. No diagrama abaixo estão elencadas as componentes em utilização, de forma direta ou indireta.

4.3.1 Diagrama de arquitetura técnica

A figura 4.2 e a tabela 4.3 abaixo descrevem a infraestrutura técnica da aplicação, com a indicação dos componentes e características de cada um.

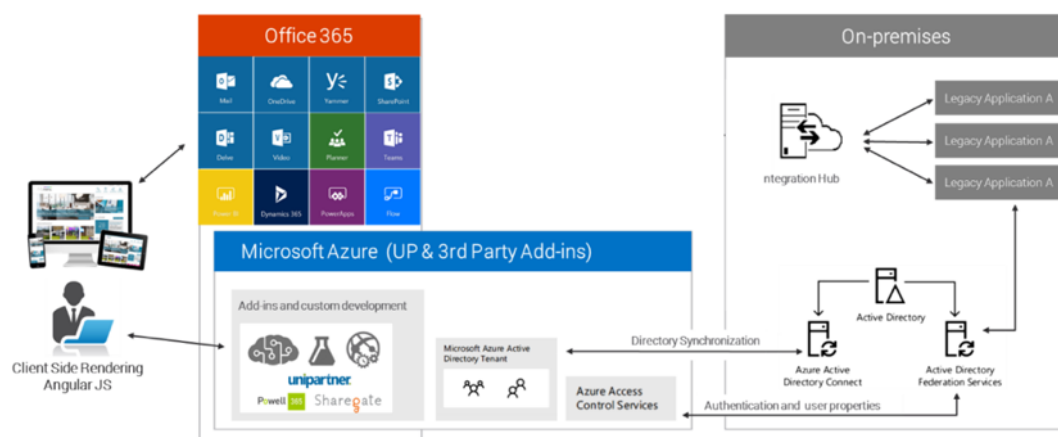


Figura 4.2: Diagrama de arquitetura técnica

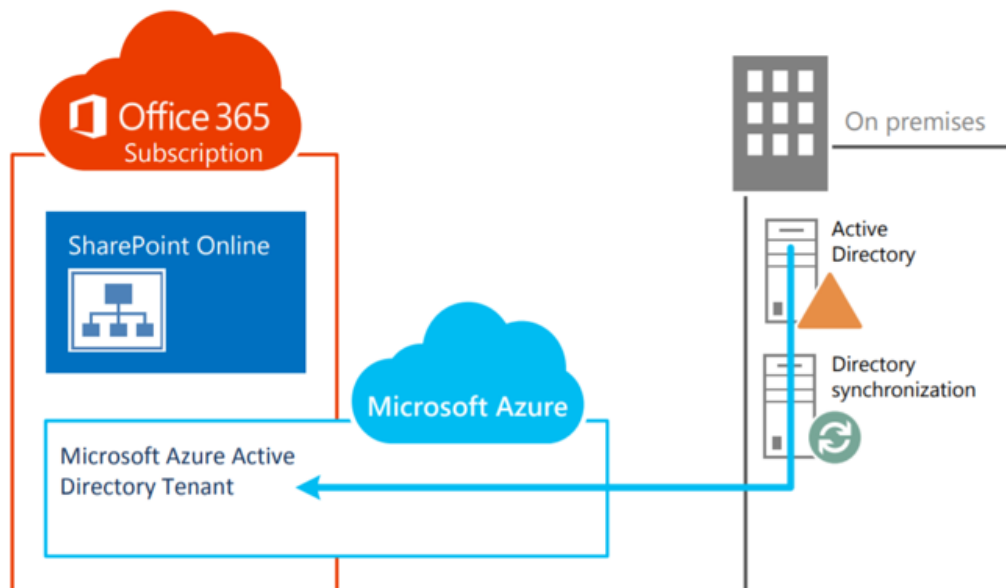
COMPONENTE	DESCRIÇÃO
<i>SharePoint On-line (Office 365)</i>	Toda a estrutura, <i>design</i> e conteúdos residem no <i>SharePoint Online</i> .
<i>Powell CDN</i>	Contém ficheiros que suportam o <i>rendering</i> de <i>web parts</i> da <i>Powell 365</i> .
<i>Graph API</i>	<i>Endpoint</i> público do <i>tenant Office 365</i> .
<i>Microsoft Azure Active Directory</i>	Toda a informação relativa a utilizadores e departamentos.

Tabela 4.3: Componentes da arquitetura técnica

Os componentes identificados na tabela 4.3, nomeadamente *SharePoint Online*, que se encontra na secção das ferramentas disponibilizadas pelo produto *Online Office 365*, na figura 4.2, é englobado em ferramentas *third-party* nomeadamente o *Powell 365*, que disponibiliza a *Powell CDN*, e a *Graph API* do *Office 365* que permite a realização de pesquisas aos conteúdos no *site*, ambos presentes na secção da *Microsoft Azure* na figura 4.2. O componente *Microsoft Azure Active Directory*, que através de sincronizações permite que as suas informações guardadas localmente fiquem disponíveis nos produtos *online* da *Microsoft*, é representado na secção *On-premises* na figura 4.2.

4.3.2 *SharePoint Online*

O *Digital Workplace* é composto por uma única *site collection* no *SharePoint Online* que foi customizada ao nível de *masterpages*, *page layouts*, *site templates*, *content types*, etc. Todo o *design*, modelo de permissões e estrutura está definido ao nível da *site collection* não existindo, para já, elementos com permissões quebradas, ou seja, herdam as permissões de *sites* superiores. A informação a apresentar está estruturada em *subsites*.

Figura 4.3: *SharePoint Online*

Na figura 4.3 pode-se observar em pormenor as sincronizações que ocorrem entre a *Active Directory* (dados locais) para o *tenant* do *Office 365* do cliente (*online*), permitindo que informações alocadas localmente possam ser disponibilizadas em serviços *online*.

Destacam-se os serviços de *User Profile* e Pesquisa que são essenciais ao bom funcionamento da solução.

User Profile

O serviço de pesquisa utiliza a informação armazenada no *SharePoint User Profile* para pesquisar pessoas.

Figura 4.4: *User Profile*

Esta informação, tal como ilustra a figura 4.4, flui a partir da AD (*Active Directory*) do

cliente (e eventualmente outros sistemas *legacy*) para a AD do *Office 365* e posteriormente para o *SharePoint UserProfile*.

Pesquisa

O serviço de pesquisa é utilizado para suportar a pesquisa dos conteúdos e das pessoas (*User Profile*)

- **Pesquisa de Conteúdos**

A utilização de uma classificação consistente normaliza a introdução de conteúdos no *SharePoint Online* contribuindo para uma maior riqueza das pesquisas produzidas. Por forma a enriquecer o serviço de pesquisa foram criados vários refinadores que permitem ter um conjunto mais alargado de filtros que podem ser aplicados a cada pesquisa.

- **Pesquisa de Pessoas**

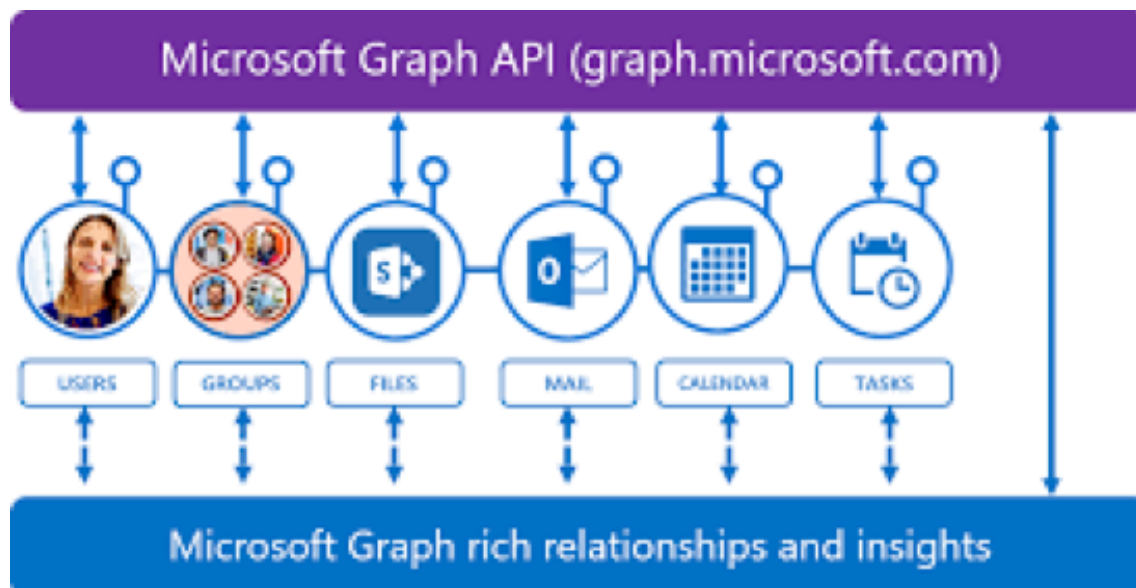
A pesquisa de pessoas consulta o *User Profile SharePoint* e está limitada à informação existente no mesmo.

- **Pesquisa de Serviços**

A pesquisa de serviços tem por base algumas listas definidas na *root* do *site*.

Graph API Office 365

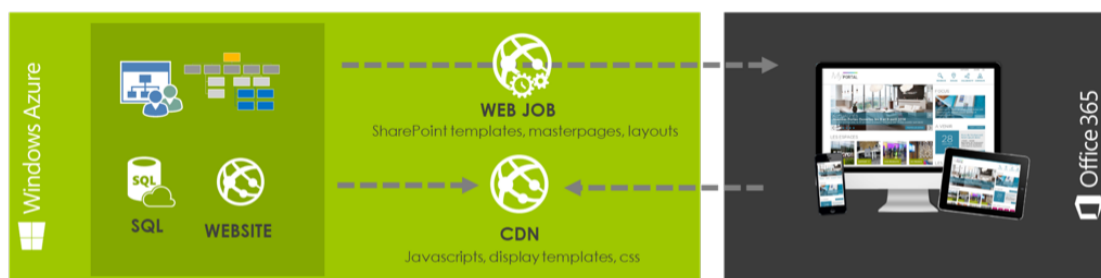
A integração com os restantes serviços de *Office 365* como *Yammer*, *One Drive*, *Office Videos*, *Mails*, *Events*, *Planner*, *Skype for Business*, entre outros, é executada através do consumo das *Office Graph APIs*, tal como ilustra a figura 4.5.

Figura 4.5: *Graph API*

As várias *web parts* existentes consomem a *Office Graph API* através de código cliente embebido nas páginas do *SharePoint Online* (ou na *CDN Powell 365*).

Powell 365

Para acelerar a implementação da solução *Digital Workplace* foi selecionado o *Powell 365* que permite configurar e gerir toda a estrutura dos *sites* desenvolvidos remotamente.

Figura 4.6: *Powell 365*

O *Powell 365* é um serviço SaaS(*Software-as-a-Service*) fornecido pela *Powel Software*. Baseado em *Azure* que facilita a configuração remota de *site collections* e *sites* em *SharePoint Online* em *tenants Office 365*, tal como ilustra a figura 4.6. O *Powell 365* é composto por dois elementos distintos: O *website Powell Manager* e a *Powell CDN*, indicados como *Web Job* e *CDN* respetivamente na figura 4.6.

Powell Manager

Ferramenta para *design* e implementação de *sites* em *SharePoint Online*, nomeadamente o *Digital Workplace*. Trata-se de um editor WYSIWYG (*What You See Is What You Get*) que suporta a configuração de *SharePoint templates*, *masterpages*, *layouts*, *content types*, etc. O *deploy* das configurações para os *tenants Office 365* é executado utilizando *web jobs* para fazer *deploy* remoto dos elementos para o *SharePoint Online*. No *Powel Manager* ficam apenas armazenadas as definições dos vários elementos de *SharePoint*, mas nunca os seus conteúdos.

Powell CDN

A *Powell CDN* utiliza a *Azure CDN Premium* para garantir redundância e disponibilidade. A *Powell CDN* é utilizada para acelerar o *rendering* dos seguintes elementos:

- Bibliotecas *Powell 365* de Javascript genéricas (AngularJS, ficheiros de Configuração *Powell 365*, livrarias JavaScripts desenvolvidas à medida, etc);
- JavaScript *web parts Powell 365*;
- AngularJS *web parts Powell 365*;
- Configuração navegação *Powell 365*;
- CSS *Powell 365*;

Estando a CDN eventualmente indisponível, o *SharePoint Online* continua a funcionar sendo apenas as *web parts Powell 365* afectadas pela indisponibilidade.

Capítulo 5

Trabalho Realizado

Neste capítulo será destacado todo o trabalho realizado ao longo do projeto, descrevendo-se de uma forma genérica os desenvolvimentos realizados a diversos componentes para os quais não foi necessário construir na íntegra as funcionalidades. De uma forma mais pormenorizada, vão ser descritos os desenvolvimentos efetuados para a realização das três funcionalidades *custom* identificadas no planeamento do projeto: duas páginas de resultados de pesquisa para colaboradores e departamentos (*peoplereults.aspx* e *servicesresults.aspx*); *web parts* que se assemelham ao estilo e funcionalidade de tabuladores (*tabs*); *web parts* que devolvem dados de taxas bancárias.

5.1 Desenvolvimentos e práticas comuns no projeto

Os desenvolvimentos de *page layouts*, a sua constituição e o seu estilo tinham como ponto de partida o desenvolvimento do seu *wireframe* por parte da *designer* da equipa. A mesma baseou-se no estilo base do produto *Powell 365* devido às condicionantes de tempo do projeto. Após desenvolver as *wireframes*, as mesmas eram validadas por nós (*developers*) de uma forma breve, de modo a verificarmos se havia algum constrangimento de desenvolvermos a página como se apresentava. As *wireframes* eram então disponibilizadas ao cliente, que posteriormente as validava ou solicitava alterações. Posteriormente, eram implementadas por nós num ambiente de desenvolvimento e evoluídas para o ambiente de qualidade, ambiente no qual o cliente validava os desenvolvimentos.

Como já foi descrito no planeamento do projeto, este foi composto por três *releases*. Antes da entrada em produção de cada *release*, foram realizadas sessões de *focus group* no ambiente de qualidade, com utilizadores-chave. Estes utilizadores realizaram vários casos de uso que englobavam as suas ações de trabalho. O *feedback* dos utilizadores serviu para se encontrar funcionalidades a melhorar e alterar as posições no *page layout* de certos elementos devido à sua maior relevância.

5.1.1 Linguagens de programação

Ao longo do projeto foi usado código HTML [10] para definir as estruturas dos *page layouts* e das *web parts*. Uma das políticas definidas desde o início do projeto foi a de reutilizar classes HTML e de criar novas classes apenas quando apresentavam especificidades no seu estilo.

Para se definir o estilo dos *page layouts* e das *web parts* foi definido código CSS [11] que afetava as classes HTML. Após o desenvolvimento dos *page layouts* e o preenchimento dos mesmos com as *web parts*, a prática comum foi de utilizar a funcionalidade de inspecionar os elementos disponível nos *web browsers*, e assim testar em tempo real os desenvolvimentos e validar se iam de encontro ao especificado nas *wireframes*. Posteriormente as propriedades CSS eram definidas no ficheiro .css de modo a ficarem disponíveis no *website*.

Por o HTML ser uma linguagem ideal para a declaração de páginas *web* estáticas, apresenta condicionantes quando se pretende realizar vistas dinâmicas, nomeadamente para o desenvolvimento das *web parts*. Foi então embebido no código HTML código AngularJS [12] para permitir a extensão do vocabulário do HTML, com a possibilidade de usufruir de várias funções, diretivas, entre outros recursos, que não existem no HTML, oferecendo assim um maior dinamismo aos componentes *web*. De modo a validar os desenvolvimentos a realizar e os valores das propriedades dos elementos, fiz uso da funcionalidade de inspecionar elementos nos *web browsers*, selecionando um determinado elemento e escrevendo na consola a seguinte expressão `angular.element($0).scope()`. A partir desta expressão conseguia analisar o elemento e fazer uso de outras funções para retirar propriedades relevantes a serem disponibilizadas nas *web parts*.

Por forma a apresentar um maior dinamismo na *intranet* realizaram-se diversas funções JavaScript [13] para ir de encontro ao comportamento esperado para certas páginas. Desenvolvi várias funções JavaScript embebidas em código HTML, através do uso das *tags* `<script>` e `</script>`. A grande maioria das funções tinham como objetivo validar se o item em questão continha um determinado valor. Para isso foi recorrente usar a chamada `document.getElementById(IDElementHTML)` e a chamada `document.getElementsByClassName(ClassElementHTML)` de modo a ter o foco no elemento *web* alvo.

Outro tipo de funções realizadas, e que tinham de ser validadas em cada página *web* da *intranet*, foram disponibilizadas num ficheiro .js que era consumido no carregamento das páginas. Estas funções validavam certos parâmetros através da seguinte expressão: `window.location().href.toLowerCase().indexOf('paginaalvo.aspx')`. Esta expressão retorna um valor inteiro positivo se a página *web* que o utilizador está a abrir é uma das páginas validadas para realizar alguma funcionalidade nessa própria página.

De modo a invocar a maioria das funções específicas para o projeto, fiz uso das

funções `setTimeout()` e `setInterval()`. Por norma, usei a função `setTimeout(funcao(), delay time)` para invocar as nossas funções, no entanto a abordagem para outras páginas com um fluxo maior de visitas teve de ser um pouco diferente. O tempo de carregamento de certas páginas sofria algumas variações consoante o número de *web parts* do tipo *searchquery* e por outras variantes como o *browser* e o número de acessos simultâneos. Para essas páginas usei a função `setInterval(funcao(), delay time)` para invocar as nossas funções, de forma a que a cada milissegundos (ms) definidos no *delay time*, a função seja invocada. Ao contrário do que se possa pensar quando o *delay time* termina, isso não é indicativo que a função começa a ser executada. Mas, que quando o *delay time* termina, uma nova mensagem (neste caso uma função) é adicionada à *queue*, e vai ser processada quando a *call stack* estiver livre para a realizar, uma vez que o código é executado assíncronamente.

5.1.2 Criações e alterações a funcionalidades da *Microsoft*

Ao longo do projeto foi recorrente ter de criar novos *content types* e realizar mapeamentos no *search schema*, por forma a designar e visualizar certas informações.

Content types

Devido à diversidade e complexidade de alguma informação do cliente teve que se criar vários *content types*, de forma a definir-se com exatidão a informação. Estes novos *content types* eram criados através da funcionalidade de clonar *content types* base disponibilizados pelo *Powell 365*, podendo-se adicionar outros campos já existentes ou então criar novos campos que herdavam diversos tipos, como por exemplo, o tipo *Date*, *Number*, *Text*, entre outros.

Estes novos *content types* criados no *Powell 365* ficavam disponíveis ao nível do *site* de *SharePoint Online* após sincronizar no *Powell 365* uma determinada estrutura, como por exemplo uma lista *SharePoint*, que tenha associado esse *content type*.

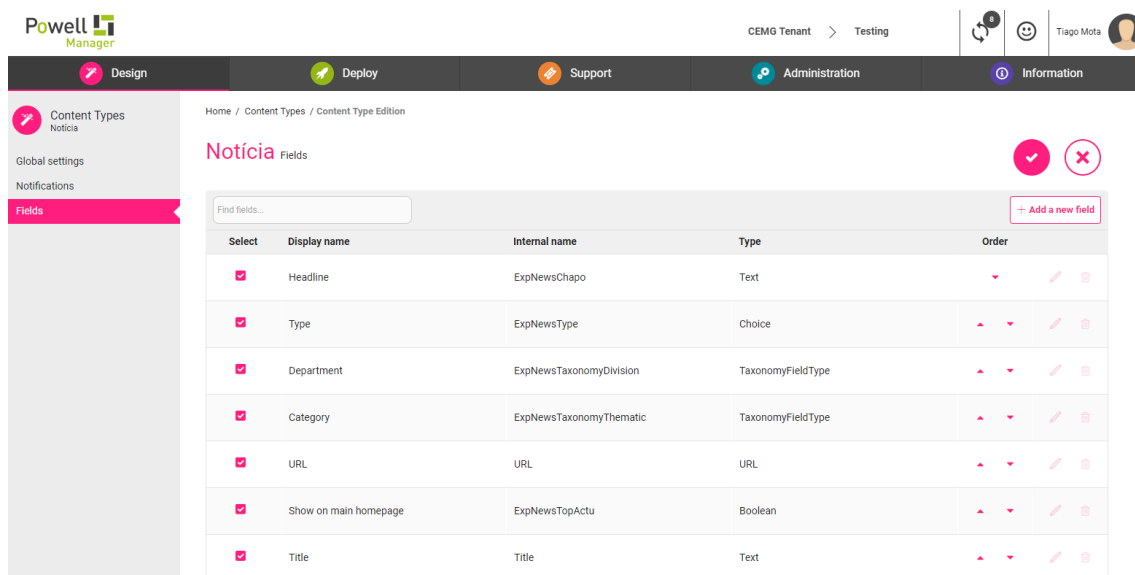


Figura 5.1: Detalhe de um *content type* criado no *Powell Manager*

Na figura 5.1 pode-se observar no *Powell Manager* os campos que fazem parte deste *content type*.

Mapeamentos no *search schema*

Ao nível do *search schema* foi necessário realizar mapeamentos a algumas propriedades nativas do *SharePoint*, dado que não é possível alterar as suas características.

Title	Texto	-	Consulta	Pesquisar	Obter	-	-	Seguro	Title, MetadataExtractorTitle, TermTitle	DocTitle, urn:schemas.microsoft.com:fulltextqueryinfo:displaytitle
-------	-------	---	----------	-----------	-------	---	---	--------	--	--

Figura 5.2: Características da propriedade *Title*

Como se pode observar na figura 5.2, as características da propriedade *Title* não são editáveis e esta propriedade não é ordenável (*sortable*), no entanto, para certas *web parts*, o requisito era que fossem ordenáveis pelo campo *Title*. Para possibilitarmos este comportamento tivemos duas opções: a primeira opção seria criar uma nova propriedade e como característica desta nova propriedade seria ser ordenável e mapeá-la para a propriedade *Title*. Posteriormente na configuração da *web part*, no campo da ordenação seria indicado o nome da propriedade que tinha sido criada; a segunda opção, que foi a aconselhada pelo serviço de suporte do *Powell 365* e pela qual se optou, era de se utilizar as propriedades disponibilizadas pelo produto *SharePoint*, as *RefinableStringXX*, *RefinableDateXX*, *RefinableIntXX*, entre outras, uma vez que estas propriedades já tinham as características necessárias para realizar o comportamento desejado. À semelhança da primeira opção, posteriormente na configuração da *web part*, no campo da ordenação seria indicado o nome da propriedade disponibilizada pelo *SharePoint*.

Foram realizados outros mapeamentos para outras situações semelhantes, onde era necessário que uma determinada propriedade tivesse a característica de ser refinável, de modo a que pudesse ser utilizada nos refinadores usados no *site*.

5.2 Resultados de pesquisa na *peopleresults.aspx* e *servicesresults.aspx*

No âmbito da primeira *release*, além da página de resultados de pesquisa predefinida do *SharePoint*, a *results.aspx*, onde são devolvidos todos os conteúdos presentes na *intranet* à exceção dos colaboradores da empresa, foi requisitada a criação de duas novas páginas que retornam resultados de pesquisa. A *peopleresults.aspx* que retorna as informações dos perfis dos colaboradores da *Active Directory* do cliente e a *servicesresults.aspx* que devolve a informação dos departamentos da empresa que se encontra disponível numa lista *SharePoint* de itens criado ao nível da *root* da *site collection* para efeitos da pesquisa em questão.

5.2.1 *Peopleresults.aspx*

Nesta página de resultados de pesquisa dos colaboradores teve-se por base a *peopleresults.aspx* disponibilizada pelo *SharePoint*. Esta é composta por várias *web parts* das quais se destaca a *web part* que devolve os resultados de pesquisa. Por defeito, esta *web part* utiliza um *search template* base, e foi a partir deste que criei um novo *search template* para ser aplicado na *web part* que retorna os colaboradores na *peopleresults.aspx*.

O cliente pretendia que fossem devolvidos mais campos *custom*, presentes no *user profile* das contas *Office 365* dos colaboradores da *Active Directory* da empresa e a remoção de outros campos que não tinham interesse para a empresa. O novo *search template* é um ficheiro HTML constituído por código HTML e por código JavaScript que, quando publicado na diretoria dos *search templates* da *site collection* gera um ficheiro JavaScript que contém o código JavaScript presente no ficheiro HTML. Para a adição de novos campos a serem apresentados, acrescentei as referências dos mesmos no cabeçalho do ficheiro HTML e acrescentei a estrutura HTML necessária. Esta estrutura HTML só fica visível se as condições efetuadas em JavaScript forem verdadeiras. Estas condições dependiam da lógica específica de negócio do cliente, mas para a maioria dos campos a lógica foi de apresentar os mesmos, se estes apresentassem valores. Para a remoção de campos excedentários, retirei a referência dos mesmos no cabeçalho do ficheiro HTML e toda a lógica presente no ficheiro para os mesmos.

As referências dos campos identificados no cabeçalho do ficheiro HTML tinham de apresentar uma correspondência no ficheiro de traduções presente na diretoria da linguagem da *site collection*. Este ficheiro é o *CustomStrings.js*, no qual acrescentei as

referências dos novos campos e as suas respetivas *labels* no seguinte formato

``Campo`` : ``Label``, como por exemplo: ``RefinableString52`` : ``Número de Colaborador``. A figura 5.3 apresenta o resultado final da *peoplereults.aspx*.

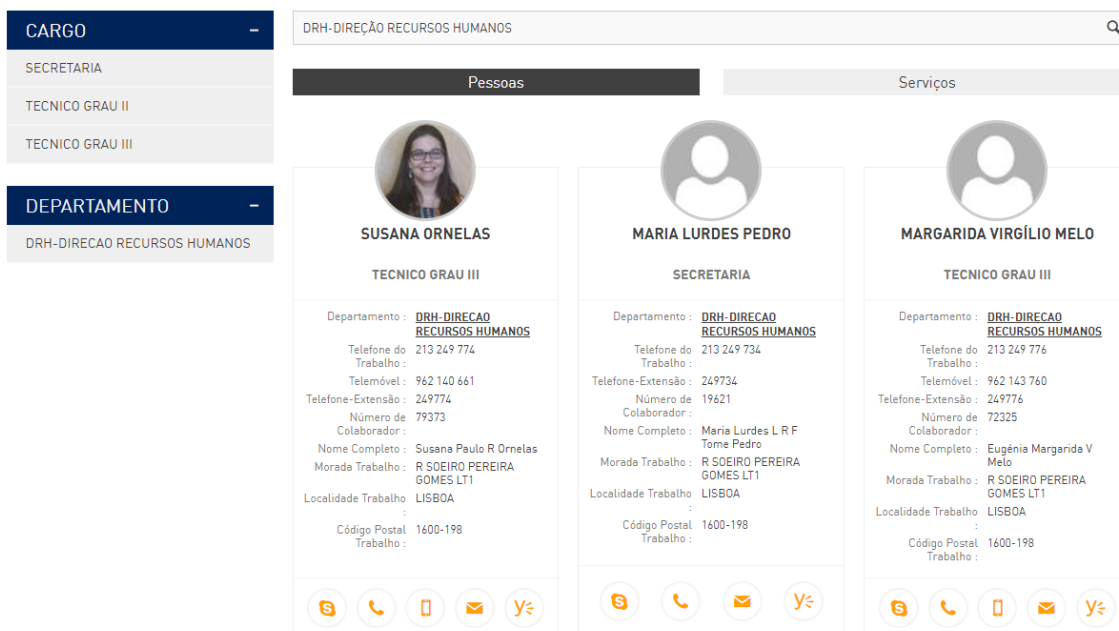


Figura 5.3: Vista da página *peoplereults.aspx*

Como se pode observar na figura 5.3, para cada colaborador é disponibilizada um leque de informações relevantes que definem o colaborador e a área de negócio a que o mesmo pertence. Defini também, uma hiperligação à página *servicesresults.aspx*, de modo a consultar a informação de um determinado departamento do colaborador que se tinha pesquisado.

5.2.2 *Servicesresults.aspx*

Para a página *servicesresults.aspx* tive por base o *search template* usado na *web part* da página de pesquisa global, criando um novo *search template* aplicado ao nível da *web part*, que devolve conteúdos apenas do *content type* criado associado à lista de departamentos. As restantes alterações foram mais profundas, face às alterações na página *peoplereults.aspx*, dado que em vez de se retornar a informação de campos *Office 365*, devolveu-se a informação presente nas colunas de uma lista *SharePoint* de itens. Esta lista é populada via *webservice*, que foi disponibilizado pelo cliente onde realiza uma pesquisa às informações dos departamentos na base de dados local do cliente e exporta as mesmas para uma lista *SharePoint* de itens que contém um *content type* que apresenta campos *custom*, criados de modo a haver um *matching* entre as colunas das tabelas da base de dados e as colunas da lista *SharePoint*.

Ao novo *search template*, que é um novo ficheiro HTML, foram adicionados ao seu corpo os campos que se pretendiam visualizar na página. Neste *search template* é desenvolvida uma variável de *output* que contém todos os campos que se pretende que sejam visíveis na página, para o ficheiro *Item_CommonItem_Body.html*.

O ficheiro *Item_CommonItem_Body.html* é consumido sempre que se abre uma página de resultados de pesquisa, e foi neste que se desenvolveu código HTML e CSS para definir a estrutura e estilo dos campos a apresentar dos departamentos e código JavaScript para validar se o contexto atual era da página *servicesresults.aspx* e se cada campo tinha valores para apresentar. A figura 5.4 apresenta o resultado final da *servicesresults.aspx*.

The screenshot displays a web application interface for searching services. On the left, there are two filter panels. The first panel, titled 'LOCALIDADE', contains a list of locations: LISBOA, AMADORA, PORTO, FUNCHAL, AVEIRO, and a 'MOSTRAR MAIS' link. The second panel, titled 'CÓDIGO-POSTAL', contains a list of postal codes: 1250-066, 1600-198, 2610-016, 2610-028, 1200-094, and a 'MOSTRAR MAIS' link. To the right of these filters is a search bar with a magnifying glass icon. Below the search bar are two tabs: 'Pessoas' (selected) and 'Serviços'. The main content area displays two search results. The first result is for 'DCC-ALMEIRIM', with a link 'Ver Pessoas' and details: Código: 0105, Tipo Departamento: Balcão, Morada: PC REPUBLICA 36, Localidade: ALMEIRIM, Código-Postal: 2080-044, Telefones: 243570550, Fax: 243570550, Feriado Municipal: 20180510, and Horário: ALMEIRIM. The second result is for 'DCC-DEP REG SANTARÉM', with a link 'Ver Pessoas' and details: Código: 1032, Tipo Departamento: Dep. Regional, Morada: LG PADRE FRANCISCO NUNES SILVA 1, Localidade: SANTAREM, Código-Postal: 2000-134, Telefones: 243001060, Fax: 243001061, Feriado Municipal: 20180319, and Horário: SANTAREM.

Figura 5.4: Vista da página *servicesresults.aspx*

Como se pode observar na figura 5.4, é possível ver as informações mais relevantes dos departamentos do cliente e uma hiperligação entre a página *servicesresults.aspx* e a página *peopleresults.aspx* para devolver todos os colaboradores de um determinado departamento selecionado pelo utilizador, através do clique na expressão “Ver Pessoas”.

5.3 Web parts de taxas bancárias

No âmbito da segunda *release*, foi planeado o desenvolvimento de *web parts custom* para devolverem valores de taxas bancárias diárias, semanais ou mensais. Estas *web parts* foram englobadas em páginas *web* que são compostas por uma secção de refinadores, de modo a que se possam filtrar resultados das taxas pela sua data.

No desenvolvimento das *web parts* das taxas bancárias, foi desenvolvido código HTML para definir a estrutura da mesma, tendo sido embebido código AngularJS para devolver os valores das taxas com um maior dinamismo, dado que os campos ao nível da *web part*

eram agregados a um vetor e era sobre este que se realizavam as pesquisas às suas propriedades. Ao nível do estilo das *web parts* foi desenvolvido código CSS, permitindo o resultado final apresentado na figura 5.5.

DIA	—	TAXAS EURIBOR				
08		08 de Maio de 2018				
MÊS	—	Euribor 360				
Maio			Last	Last1	Last2	Last3
ANO	—	1Y	2.00000	3.00000	4.00000	5.00000
2018		2M	1.00000	23.0000	4.00000	6.00000
		Datas	2018-06-14 .	2018-06-15 .	2018-06-16 .	2018-06-17 .

Figura 5.5: Página das taxas Euribor

Como se pode observar na figura 5.5, para além da *web part* a página é composta por uma secção de refinadores, que apresenta como refinadores o dia, o mês e o ano das taxas Euribor. Estes valores foram gerados a partir de *calculated columns* (descritas mais à frente). Outro desenvolvimento relevante nesta página foi a realização de mapeamentos no *search schema* para devolver as propriedades que são usadas nos refinadores.

5.3.1 *Calculated columns*

Para as taxas bancárias houve a necessidade de dividir uma data, presente numa coluna de uma lista de *SharePoint*, em três colunas *SharePoint* do tipo *calculated column*. Cada uma destas novas colunas contém uma expressão que permite devolver os resultados esperados, como por exemplo: `=TEXT([Date];`dd'`)`, que devolve o apenas o dia da data. Tal como se pode observar na figura 5.6 a expressão que devolve o dia de uma data.

Definições ▸ Editar coluna ⓘ

Nome e tipo

Nome da coluna:

Tipo de informações desta coluna: Calculado (cálculo baseado noutras colunas)

Definições de Coluna Adicionais

Descrição:

Fórmula:

Inserir Coluna:

- Criado
- Data
- Descricao
- Hora
- ID de Recurso de Con...
- Meses
- Modificado
- MonthCalculated
- OrderCalculated
- Taxa

Adicionar à fórmula

Tipo de dados devolvido a partir desta fórmula:

- ☒ Uma linha de texto
- ☐ Número (1 / 1,0 / 100)
- ☐ Moeda (\$, €, £)
- ☐ Data e Hora
- ☐ Sim/Não

Figura 5.6: Coluna *SharePoint* que devolve o dia de uma data do tipo *calculated column*

Criei outra coluna *SharePoint* do tipo de *calculated column* para que a ordenação da apresentação de resultados fosse sobre essa coluna. A expressão semelhante que é usada nessa coluna é a seguinte: `=IF([Title]='1NomePretendido', '1', IF([Title]='2NomePretendido', '2', ''))`. Valida-se se o valor presente na coluna *Title* é o “1NomePretendido”, se for devolve o número 1, senão continua a validar os outros possíveis valores na coluna *Title*, atribuindo-lhes outros números distintos.

5.3.2 Mapeamentos de propriedades no *search schema*

Ao nível do *search schema* foi necessário realizar alguns mapeamentos às colunas do tipo *calculated column* uma vez que estas novas colunas não tinham a característica refinável (*refinable*). Como tal, realizaram-se mapeamentos para propriedades que tinham essa característica para as colunas do tipo *calculated column*. Após os mapeamentos, nos refinadores indicaram-se as propriedades mapeadas atingindo-se o comportamento espetável e visível na figura 5.5.

5.4 Web parts do estilo de tabuladores

No âmbito da última *release*, para uma determinada área de negócio o cliente pretendia um aspeto visual inovador para certas *web parts* que não existia no produto *Powell 365*. Pretendiam um estilo de *tabs* para apresentar informações que se encontravam organizadas em categorias e sub-categorias.

Como tal, e à semelhança da construção de outras *web parts*, foi definido código

HTML e CSS para estruturar e definir o estilo da *web part*, assim como foi embebido código AngularJS para validar e apresentar o comportamento esperado.



Figura 5.7: *Web part Tabs*

Na *web part* (figura 5.7) fez-se uso das diretivas `ng-class` e `ng-click` do AngularJS. Para a definição de uma determinada *tab*, isto é, uma categoria do negócio, realizou-se este pedaço de código:

```
<div class='`tabs-list`'> <li class='`tabs-list-item ng-scope`'  
data-ng-click='`vm.tab = 'Oferta Particulares`''  
data-ng-class='`'active': vm.tab === 'Oferta Particulares`''`>  
<a class='`ng-binding`' id='`Tab1`'>OFERTA PARTICULARES</a> </li>  
</div>
```

onde se cria uma lista de *tabs*, onde cada *list item* (``) é uma *tab*.

Para cada *tab*, `<li class='`tabs-list-item ng-scope`'>`, foram definidas outras diretivas. A diretiva `data-ng-class='`'active': vm.tab === 'Oferta Particulares`''`` serve para adicionar à classe HTML a propriedade “active”, de modo a alterar as propriedades CSS, destacando a *tab* como a que está ativa. A diretiva `data-ng-click='`vm.tab = 'Oferta Particulares`''``, serve para, ao clicar naquele elemento HTML, se desenrolar uma ação, que está definida na restante *web part*. Ao clicar-se numa categoria irão ser disponibilizadas as suas subcategorias, isto é, as suas *subtabs*, que quando selecionadas mostram os conteúdos que lhe estão associados.

Capítulo 6

Discussão

Neste tópico de discussão sobre o projeto em estudo abordo alguns pontos que mereceram ponderação entre várias possíveis soluções, em que a decisão incluía tanto a equipa de desenvolvimento como o cliente e a equipa de suporte do produto usado. Indico as abordagens praticadas ao longo do projeto, corroborando-as, ou não, como as melhores práticas no desenvolvimento de projetos que usam o produto *SharePoint Online*.

6.1 *Performance*

A realização de boas práticas foi sempre um dos pontos-chave das nossas ambições enquanto equipa de desenvolvimento. Tendo em conta que o projeto realizado em *SharePoint Online* foi desenvolvido apenas em *client-side*, teve-se algumas das precauções sugeridas na documentação oficial do produto [14].

A sequência de acontecimentos de *sites* modernos *client-side* é a seguinte:

Primeira visita da página:

- Solicitação de página;
- Emitir solicitações de arquivos de recursos (zero ou mais);
- Execução de algum *JavaScript*;
- Emitir solicitações de dados (zero ou mais);
- Execução de mais *JavaScript*.

Visitas seguintes à página:

- Solicitação de página;
- Execução de algum *JavaScript*;
- Emitir solicitações de dados (zero ou mais);

- Execução de mais *JavaScript*.

Como se pode observar nos itens identificados acima, existe uma grande dependência da execução de código *JavaScript*, contribuindo significativamente para o bom ou mau desempenho de uma aplicação desenvolvida em *client-side*.

As boas práticas de desenvolvimento em *client-side* indicam que se use um modo de telemetria para quantificar os possíveis problemas de *performance*, sendo essencial obter métricas objetivas da *intranet*. Como tal, incorporámos no código da *masterpage* do *site* o módulo de código *JavaScript* fornecido pelo uso do *Azure Application Insights* para monitorizar a *performance* em tempo de execução da *intranet*. Havia a possibilidade de realizar o nosso sistema de telemetria, no entanto o mesmo não foi desenvolvido por não se encontrar dentro do âmbito do projeto e por não ser recomendado pelo produto, dado que poderia prejudicar a *performance* do *site*.

Aquando do desenvolvimento dos requisitos da primeira *release*, não foram detetados problemas relevantes de *performance*, assim como no *focus group* referente a esta *release*. Após a entrada em produção da primeira *release*, o número de utilizadores era consideravelmente maior, face à amostra de teste no ambiente de qualidade, tendo-se levantado então a problemática da *performance*. Alguns colaboradores do cliente, presentes em espaços físicos diferentes, passavam por períodos duradouros de espera de acesso à nova *intranet*, em média cerca de 6 a 7 segundos. Na navegação entre as páginas da *intranet*, o tempo médio de resposta era de cerca 2 a 3 segundos. Várias possíveis abordagens foram levantadas pelos intervenientes do projeto para tentar dar resposta a este problema: aumentar a velocidade de *internet* nos locais de trabalho onde havia um maior número de queixas; mudar-se o *browser* oficial do cliente, que é o *Internet Explorer* versão 11, para um *browser* mais atual e rápido; alterar-se o conteúdo de algumas páginas da *intranet* que tinham um grande número de acessos, reduzindo-se o número de *web parts* na página, ou alterar-se algum outro procedimento feito pela equipa de desenvolvimento que estivesse a despoletar estes tempos de acesso; solicitar alterações ao próprio produto por parte da equipa de suporte da *Powell 365*.

Aumentar a velocidade da *internet*

Sobre a possível solução de se aumentar a velocidade de *internet* nos locais de trabalho mais afetados, o cliente realizou testes de velocidade da *internet* e chegou-se à conclusão de que a velocidade de *internet* não era igual em todos os postos de trabalho, mas que não seria a principal razão para estes tempos de carregamento por parte da nova *intranet*.

Alteração do *browser*

Outra possível alteração a realizar-se no projeto, seria a alteração do *browser* oficial da *intranet* para um *browser* mais atual e rápido como é o caso do *Google Chrome*. É comum

que uma empresa de grande dimensão tenha aplicativos que exijam o uso de um *browser legacy* [14]. No entanto, essa restrição não deve impedir o avanço dos novos *websites*, devendo-se projetar novos aplicativos que tirem partido das funcionalidades dos *browsers* modernos.

Esta sugestão foi feita inicialmente pela nossa equipa de desenvolvimento por validarmos que os tempos de resposta do *Google Chrome* seriam melhores em cerca de 0,5 segundos face aos mesmos acessos quando realizados no *Internet Explorer* versão 11. Posteriormente esta mesma ação foi sugerida oficialmente pela equipa de suporte da *Powell 365* ao nosso cliente que insistia em ter uma resposta oficial de solução por parte do produto.

A sugestão foi rejeitada oficialmente pelo cliente, uma vez que os colaboradores mais afetados trabalham em *virtual machines* (VMs). No passado as VMs apresentavam um espaço de disco e memória RAM já reduzidos e com este novo projeto o cliente decidiu manter a mesma abordagem, por não sentir a necessidade de aumentar o número de servidores para este projeto. Face a este paradigma de os colaboradores trabalharem em VMs e noutras aplicações *legacy* que só são executadas no *browser Internet Explorer*, o cliente rejeitou instalar por exemplo, o *browser Google Chrome*, em todas as VMs. Para os trabalhadores do cliente que não trabalham em VMs, foi instalado o *browser Google Chrome* e foi indicado que usassem preferencialmente este *browser* para aceder à nova *intranet*. Independentemente do *browser* que se usasse para se aceder à *intranet*, os tempos de resposta continuavam elevados face ao requisitado pelo cliente.

Otimização dos desenvolvimentos

Uma das outras soluções consistia em otimizar ao máximo os nossos desenvolvimentos principalmente nas funções *JavaScript*, de modo a não termos grande impacto no tempo de carregamento das páginas. Evitámos, então, realizar funções *JavaScript* e código CSS redundantes que eram invocados aquando do carregamento das páginas *web*, sendo importante também realçar que o próprio produto *Powell 365* incorpora código *core JavaScript*. Solicitou-se também ao cliente que, em termos de conteúdo, não carregasse imagens de tamanho superior a 10 *megabytes* (MB).

As *web parts* tinham a possibilidade de guardar *cache* dos resultados devolvidos na mesma e tirámos partido dessa funcionalidade. Os resultados presentes em *cache* eram devolvidos se estivessem presentes e se ainda não tivessem expirado, isto é, se houvesse um *cache hit*.

Após vários testes de *performance* com e sem as nossas *web parts* e com e sem as nossas funções *JavaScript*, verificou-se que as diferenças de tempo de resposta por parte das páginas *web* não eram significativas. No entanto, o número de *web parts* presente numa página tinha implicações na variação do tempo de resposta. Reportámos este facto ao suporte da *Powell 365* e foi confirmado que existiam variações de tempo de resposta

para páginas *web*, cerca de 0,5 segundos, se fossem compostas por mais de 6 *web parts*. A equipa de suporte do produto suavizou o problema através de alterações estruturais nas invocações das *web parts*, e optaram por não divulgar claramente quais foram as suas otimizações ao nível das *web parts*.

Otimização do produto

A última proposta de solução, proposta pelo cliente, foi a possível otimização de processos internos do produto por parte da equipa de suporte da *Powell 365*.

O produto faz uso da *Content Delivery Network* (CDN) [14]. A CDN é uma rede dispersa geograficamente que permite que o utilizador final obtenha um determinado recurso da CDN mais próxima. O uso de um CDN resulta em melhores tempos de *download* e contribui para uma melhor perceção do desempenho geral das páginas *web*. A *Powell 365* aproveitou a existência das CDNs para fornecer estruturas *third-party client-side*, como por exemplo desenvolvimentos em *JQuery*, *AJAX* e *JavaScript*.

Após alguns testes validou-se desde cedo que o tempo de resposta por parte da CDN da *Powell 365* era elevado face ao esperado e que parecia ser o principal responsável pelo tempos demorados de resposta. A questão foi então levantada ao suporte da *Powell 365* e realizaram melhorias no seu código *AJAX* e *JavaScript*, não tendo divulgado em concreto as melhorias que realizaram. Os tempos de resposta melhoraram em cerca de 1 a 1,5 segundos, isto é, o tempo de acesso à página inicial da intranet passou para cerca de 5 a 6 segundos e 1,5 segundos na navegação entre páginas no *browser* oficial do cliente.

Apesar desta melhoria, o cliente não ficou totalmente satisfeito querendo uma melhor otimização; no entanto a *Powell 365* rejeitou fazer alterações mais profundas ao seu código *core*, dado que em *browsers* modernos estava a obter resultados de 3 a 4 segundos no acesso à página inicial e 1 segundo na navegação entre páginas. Estes valores eram aceitáveis por parte do cliente, visto serem valores próximos dos registados na *intranet* antiga. A *Powell 365* sugeriu então ao nosso cliente que passasse a utilizar *browsers* modernos para o uso deste *website*.

6.2 Mapeamentos no *Search schema*

Como já foi referido anteriormente, as alterações no *search schema* ao nível do *tenant* do cliente são replicadas para os *search schemas* das *site collections* criadas no *tenant*. No caso deste cliente e de outros clientes que utilizem o produto *SharePoint*, os mesmos podem construir vários *websites* para diferentes áreas do negócio, apresentando cada *website* as suas especificidades.

Para este projeto, e após sugestão da equipa de suporte da *Powell 365*, realizámos mapeamentos de algumas propriedades que não tinham todas as características necessárias para o funcionamento pretendido pelo cliente, essencialmente para que as propriedades

pudessem ser ordenáveis e/ou pesquisáveis (*Sortable, Searchable*), para as propriedades já disponibilizadas pelo produto *SharePoint* do tipo *RefinableStringXX*, e para outros tipos de dados.

Estes mapeamentos foram realizados ao nível do *tenant* de modo a serem replicados ao nível das *site collections*; obviamente que os mesmos podiam ser realizados ao nível de cada *site collection*, mas o cliente não preferiu esta abordagem por ser mais suscetível ao erro humano (neste caso de trocar o mapeamento de uma determinada propriedade).

Independentemente do local onde eram realizados os mapeamentos, algo que poderíamos ter feito de diferente era criar novas propriedades com as características requeridas e mapear para estas novas propriedades as propriedades que pretendíamos devolver nas *web parts*. Estas novas propriedades teriam o desígnio de *intranet*, de modo a que fossem facilmente identificáveis no *search schema* do *tenant* e para que *site collection* haviam sido criadas. A única desvantagem da criação destas novas propriedades era o tempo de indexação do produto que poderia levar até 48 horas e se após este período as novas propriedades não continuassem disponíveis o cliente teria de abrir um *ticket* à *Microsoft* a solicitar um *full crawl* e teria de justificar internamente a abertura do mesmo.

Face a este problema o cliente decidiu enveredar pelo processo mais simples que era o de reutilizar as propriedades disponibilizadas pelo produto. Se a escolha fosse tomada pela equipa de desenvolvimento, ter-se-ia enveredado pela criação de novas propriedades por uma questão de organização interna do projeto.

6.3 Aprovação documental

Para certos tipos de documentos internos e externos de grande relevância para o cliente e que são resultantes do trabalho colaborativo entre os seus trabalhadores, optou-se por usar funcionalidades *out-of-the-box* do produto *SharePoint* de aprovação documental e de processos de *check-in* e *check-out* de ficheiros.

No caso da funcionalidade de aprovação documental, após a criação dos documentos por parte dos colaboradores pertencentes ao grupo *SharePoint* de editores, os documentos ficam no estado pendente até serem aprovados ou rejeitados por um outro grupo de colaboradores que pertencem ao grupo *SharePoint* de aprovadores. Quando os documentos ficavam com o estado de aprovação ficavam visíveis para os restantes grupos de *SharePoint* da *site collection*.

A funcionalidade de *check-out* de um documento permite que apenas um colaborador possa editar o documento, impedindo que vários colaboradores possam fazer alterações ao mesmo tempo, o que pode levar a conflitos de versões dos documentos. Quando o colaborador tiver terminado as suas alterações pode libertar o documento, fazendo o *check-in* do mesmo, para que mais colaboradores possam realizar mais alterações ao documento a partir das efetuadas anteriormente.

Estas duas funcionalidades tiveram uma grande importância no trabalho colaborativo entre diferentes áreas de responsabilidades do cliente.

6.4 Modelo de permissões

A organização da estrutura do *site*, nomeadamente a criação de *subsites*, foi definida consoante as áreas de negócio do cliente, de modo a se poder designar permissões específicas a certos *subsites* e aos seus *subsites*. Por defeito, todos os *subsites* herdam as permissões do seu *site* superior, no entanto as permissões podem ser quebradas e definir-se permissões específicas.

Para alguns *subsites* da *intranet* quebrámos as permissões dado que existiam utilizadores pertencentes a grupos de *SharePoint* que tinham de ter permissões de leitura do *subsite* e que não tinham permissões de leitura do *subsite* superior. Estas permissões de leitura permitem aos utilizadores terem acesso de leitura a todas as páginas do *subsite*, no entanto pode haver *web parts* que não devem estar visíveis para certos grupos e/ou utilizadores. Para tal, fez-se uso de outra funcionalidade *out-of-the-box* do produto *SharePoint* que consiste em atribuir audiências às *web parts* de certas páginas *web*. As audiências indicam quais os grupos e/ou utilizadores que conseguem visualizar a *web part*.

Este modelo permitiu aos utilizadores terem acesso apenas aos conteúdos necessários para a sua atividade laboral na *intranet*. Este foi um dos pontos críticos apontados pelo cliente devido ao acesso à *intranet* de colaboradores externos, e de estes só poderem ter acesso aos conteúdos direcionados aos mesmos.

6.5 Alternativas ao projeto e complementares

A conjuntura temporal do projeto não permitiu derivar para outras possíveis abordagens de projetos realizados em *SharePoint* com um maior nível de customização e de desenvolvimento, dado ter sido um projeto fechado e de curta duração. Poder-se-iam ter integrado mais componentes e um maior número de novas funcionalidades, no entanto, os gestores de projeto mostraram-se intransigentes e, apesar de aceitarem as sugestões da equipa de desenvolvimento, algumas destas teriam de ser rejeitadas face às limitações negociadas no projeto.

Capítulo 7

Conclusão

Com a renovação do *site* interno do cliente para a versão *online* do *SharePoint*, conseguiu-se otimizar a aquisição e partilha de conhecimento entre os colaboradores, num ambiente mais interativo e corporativo, que tenderá a resultar numa possível maior produtividade interna.

As funcionalidades *out-of-the-box* do produto *SharePoint* iam de encontro às necessidades do cliente, e tal como foi definido no âmbito do projeto, fez-se uso das mesmas. Para as funcionalidades *out-of-the-box*, nomeadamente *web parts template*, foram realizados desenvolvimentos sobre as mesmas, de modo a que a informação disponibilizada pelas mesmas fosse de encontro aos requisitos.

Para além destas funcionalidades, o âmbito do projeto abrangia o desenvolvimento integral de certas funcionalidades como um sistema de pesquisa que retorne pessoas e departamentos da empresa, o desenvolvimento de *web parts* que devolvam os valores de certas taxas de mercado e o desenvolvimento de *web parts* com o aspeto de tabuladores, inexistentes no produto.

A concretização dos vários objetivos do projeto que dependiam da equipa de desenvolvimento foram atingidos. Teve-se o cuidado de não sobrecarregar a *intranet* com código cliente, assim como se desenvolveram todas as funcionalidades e *layouts* abrangidos no âmbito do projeto. O tratamento da informação existente e a facilidade da gestão de novos conteúdos na *intranet* foi ajustada ao cliente, facilitando a interação do mesmo com este novo portal.

Os objetivos que não dependiam da equipa de desenvolvimento, nomeadamente a lentidão do *core* de alguns produtos da *Microsoft* usados e as limitações do projeto para o desenvolvimento de funcionalidades *custom*, fizeram com que a *intranet* se assemelhasse a uma *intranet* moderna e não a um *website* externo moderno como era a ambição do cliente.

No entanto, após o *feedback* dos seus utilizadores, pode-se afirmar que houve um *upgrade* no envolvimento tecnológico do cliente quando comparado com a antiga *intranet* desenvolvida em *SharePoint* 2007.

7.1 Comentários críticos

7.1.1 Ao produto

Face ao âmbito do projeto ser baseado praticamente nas funcionalidades *out-of-the-box* dos produtos *Microsoft*, o *layout* das informações e o modo de introduzir novos conteúdos eram os espectáveis no ambiente disponibilizado pelas ferramentas da *Microsoft*.

Por o projeto apresentar esta vertente, o mesmo ficou limitado a desenvolvimentos customizados. Existem diversos exemplos de *websites* internos construídos sobre *SharePoint Online* [15] (figura 7.1) que apresentam *layouts* modernos que fazem desvanecer o estigma do seu *layout* não fornecer uma rica *user experience*, através do uso de outras tecnologias complementares.

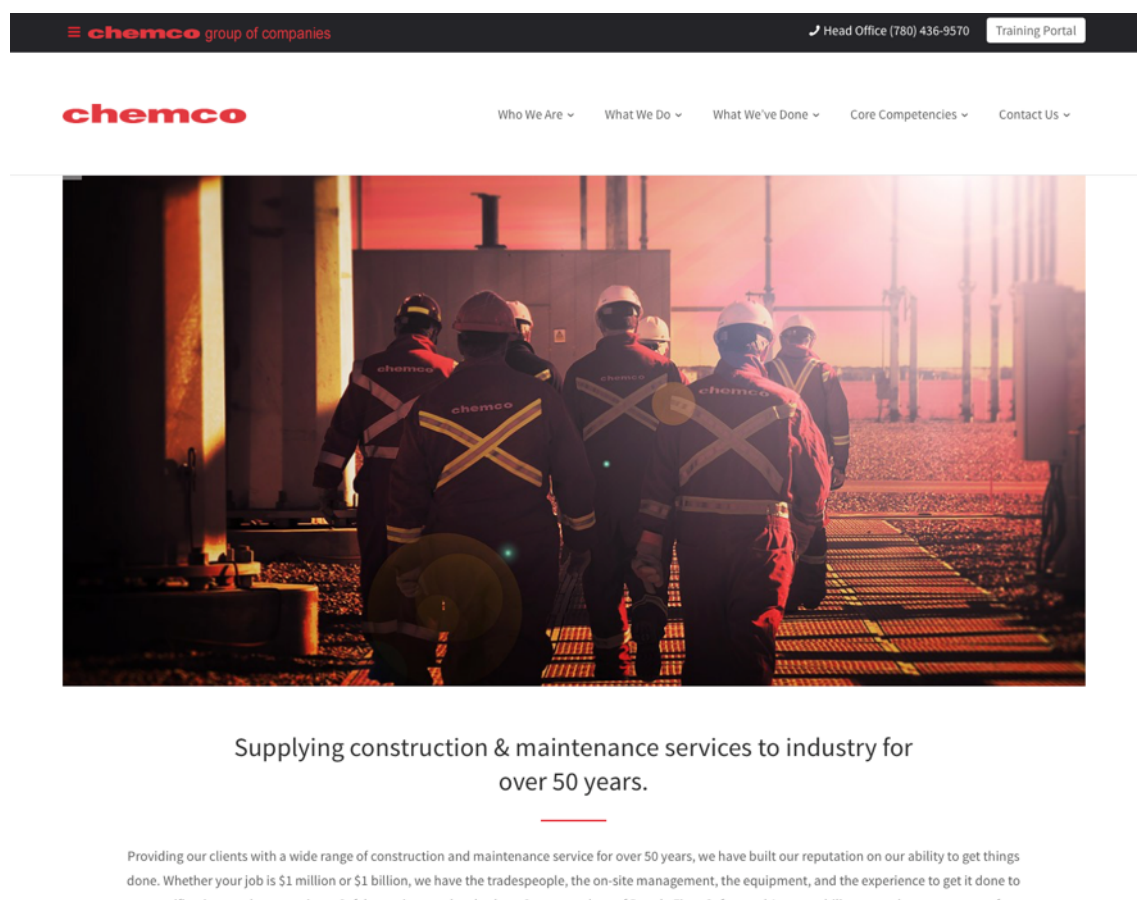


Figura 7.1: Exemplo de *website* desenvolvido em *SharePoint Online* para um empresa na área da construção civil

Como já foi descrito no capítulo das tecnologias usadas, o tempo da indexação de conteúdos na *intranet* construída em *SharePoint Online* foi uma das queixas do cliente. Um exemplo prático deste comportamento de indexação é o seguinte: após a introdução de uma nova notícia interna do cliente da Unipartner, a mesma só ficará disponível quando

a indexação ocorrer, sendo que a indexação só ocorrerá depois do *continuous crawler* validar que existe um novo conteúdo a indexar, demorando cerca de 15 minutos. O cliente, para confirmar que os conteúdos introduzidos estão corretos e que aparecem no local suposto da *intranet*, tem de esperar pelo *continuous crawler*. Como se trata de uma funcionalidade *out-of-the-box* do produto não é possível realizar qualquer alteração com o intuito de diminuir o tempo de indexação de conteúdos. Como esta queixa é comum à maioria dos clientes que usam a ferramenta *SharePoint Online*, a *Microsoft* está a otimizar o *continuous crawler*.

7.1.2 Ao projeto

Apesar da melhoria efetiva obtida no portal interno do cliente face ao anterior, se o mesmo fosse integrado com outras tecnologias ou se no planeamento do mesmo houvesse uma componente específica para o melhoramento genérico do *layout* do *website*, certamente que os poucos *feedbacks* menos positivos, que se referiam ao facto de o *website* se assemelhar a um *website* interno e não a um *website* externo como era o idealizado pelo cliente, já não iriam existir.

Na minha opinião, como foi referido neste relatório, a conjuntura do mesmo limitou a possibilidade de se realizar um maior número de desenvolvimentos *custom*, com os quais poderia ter evoluído mais significativamente como *front-end developer* face aos possíveis novos desafios com que me iria deparar. Apesar deste fator determinante, sinto que desenvolvi novas *skills* enquanto *front-end developer*, mas gostaria de ter tido a possibilidade de realizar mais desenvolvimentos sem ter *templates* que já tinham grande parte do trabalho feito.

7.2 Trabalho futuro

Como trabalho futuro, fruto do âmbito do projeto, não existem grandes novos desenvolvimentos que se possam realizar, além de certos melhoramentos: como criar novamente certos *content types* que têm *fields* em excesso que não são relevantes para o paradigma do cliente ao invés de os ocultar nas listas de *SharePoint*; a reorganização de certos conteúdos de modo a ser mais intuitivo para os editores de conteúdo encontrarem as listas de *SharePoint* respetivas; outras pequenas alterações que se enquadrem dentro do âmbito do projeto.

Porém, se não houvesse esta limitação no projeto, poder-se-iam incorporar novos desenvolvimentos *custom*, como por exemplo novas *web parts* que iriam trazer um maior dinamismo e novidade à informação apresentada, assim como outras tecnologias *Microsoft*, como por exemplo *Power BI* [16] que permite uma melhor gestão dos dados analíticos das empresas, de modo a enriquecer a *intranet*.

7.3 Alternativas ao *Powell 365*

Para outros projetos desenvolvidos em *SharePoint Online*, ao invés do uso da ferramenta *Powell 365* como complementar do *Sharepoint Online* para melhorar o *layout* do *website*, existem muitas mais opções no mercado das *intranets* corporativas integradas no produto *Office 365*, dos quais destaco duas ferramentas, a *Valo* [17] e a *PRYSM* [18] que juntamente com o *Powell 365* têm as melhoras críticas dos utilizadores e *developers*.

A escolha de qual ferramenta se deverá usar para o desenvolvimento de *intranets* colaborativas deverá ser consoante decisões internas por parte dos gestores de projeto em termos financeiros e em termos de rapidez de desenvolvimento e a realidade e a área de negócio do cliente, de modo a que o produto final seja similar ao idealizado pelo cliente.

Bibliografia

- [1] Shaifali Gupta Charul Nigam. “Agile Methodology for Software Development”. Em: *IITM Journal of Information Technology* 3 (2017), pp. 54–64. DOI: <http://iitmjp.ac.in/wp-content/uploads/2015/11/IITM-Journal-2017.pdf>.
- [2] Nabil Sultan. “Knowledge management in the age of cloud computing and Web 2.0: Experiencing the power of disruptive innovations”. Em: *International Journal of Information Management* 33 (2013), pp. 160–165. DOI: <https://www.sciencedirect.com/science/article/pii/S0268401212001041>.
- [3] Microsoft. *Introduction to content types and content type publishing*. URL: <https://support.office.com/en-us/article/introduction-to-content-types-and-content-type-publishing-a5026d23-8df8-42f6-b0d6-1920880c0d03> (acedido em 05/01/2018).
- [4] Johnny Tordgeman. *Learning Search-driven Application Development with SharePoint 2013*. Birmingham, UK: Packt Publishing Ltd, 2013.
- [5] Microsoft. *Gerir o Centro de Pesquisa do SharePoint Online*. URL: <https://support.office.com/pt-pt/article/gerir-o-centro-de-pesquisa-do-sharepoint-online-174d36e0-2f85-461a-ad9a-8b3f434a4213?ui=pt-PT&rs=pt-PT&ad=PT> (acedido em 07/01/2018).
- [6] Microsoft. *Manage the search schema in SharePoint Online*. URL: <https://support.office.com/en-us/article/manage-the-search-schema-in-sharepoint-online-d4fab46d-ba41-4c03-9d4c-32b5b33198b6> (acedido em 10/01/2018).
- [7] Microsoft. *About multiple online instances or tenants*. URL: <https://docs.microsoft.com/en-us/dynamics365/customer-engagement/admin/multiple-online-instances-tenants> (acedido em 27/03/2018).
- [8] Microsoft. *SharePoint Online planning guide*. URL: <https://support.office.com/en-us/article/sharepoint-online-planning-guide-abacd1bb-295d-4235-afdd-15f5e4cc2e6c?ui=en-US&rs=en-US&ad=US> (acedido em 31/03/2018).
- [9] Microsoft. *Create and use site templates*. URL: <https://support.office.com/en-us/article/create-and-use-site-templates-60371b0f-00e0-4c49-a844-34759ebdd989> (acedido em 02/04/2018).
- [10] Wikipedia. *HTML*. URL: <https://en.wikipedia.org/wiki/HTML> (acedido em 21/10/2017).

- [11] Wikipedia. *Cascading Style Sheets*. URL: https://en.wikipedia.org/wiki/Cascading_Style_Sheets (acedido em 21/10/2017).
- [12] AngularJS. *AngularJS*. URL: <https://angularjs.org> (acedido em 20/02/2018).
- [13] Techopedia Inc. *JavaScript (JS)*. URL: <https://www.techopedia.com/definition/3929/javascript-js> (acedido em 21/10/2017).
- [14] Microsoft. *Performance guidance for SharePoint Online portals*. URL: <https://docs.microsoft.com/en-us/sharepoint/dev/solution-guidance/portal-performance> (acedido em 14/03/2018).
- [15] chemco. *Website chemco*. URL: <https://www.chemco.com/> (acedido em 08/12/2017).
- [16] Microsoft. *Website Microsoft Power BI*. URL: <https://powerbi.microsoft.com/en-us/> (acedido em 20/05/2018).
- [17] Valo. *Website Valo*. URL: <https://www.valointranet.com/> (acedido em 20/05/2018).
- [18] PRYSM. *Website PRYSM*. URL: <https://www.prysm.com/resources/library/microsoft-online-quick-guide/> (acedido em 20/05/2018).